



Contributions à la modélisation mathématique et à l'algorithmique parallèle pour l'optimisation d'un propagateur d'ondes élastiques en milieu anisotrope

Lionel Boillot

► To cite this version:

Lionel Boillot. Contributions à la modélisation mathématique et à l'algorithmique parallèle pour l'optimisation d'un propagateur d'ondes élastiques en milieu anisotrope. Modélisation et simulation. Université de Pau et des Pays de l'Adour, 2014. Français. NNT: . tel-01133713v2

HAL Id: tel-01133713

<https://inria.hal.science/tel-01133713v2>

Submitted on 12 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT

Présentée par

Lionel BOILLOT

Pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR

À

L'Université de Pau et des Pays de l'Adour

École Doctorale des Sciences et leurs Applications - ED 211

DOMAINE DE RECHERCHE : Mathématiques Appliquées

**Contributions à la modélisation mathématique
et à l'algorithmique parallèle pour l'optimisation
d'un propagateur d'ondes élastiques en milieu anisotrope**

Directrice de Thèse : **Mme Hélène BARUCQ**

Co-direction : **M. Julien DIAZ, M. Henri CALANDRA**

Soutenue le 12 décembre 2014

Après avis de :

Mme Laurence HALPERN	Professeur, Université Paris 13	Rapporteur
M. Frédéric DESPREZ	Directeur de Recherche, Inria, ENS Lyon	Rapporteur

Devant la commission d'examen formée de :

M. Emmanuel AGULLO	Chargé de Recherche, Inria	Examineur
Mme Hélène BARUCQ	Directrice de Recherche, Inria	Directrice de thèse
M. Henri CALANDRA	Ingénieur de Recherche, Total E&P	Co-directeur de thèse
M. Julien DIAZ	Chargé de Recherche, Inria	Co-directeur de thèse
M. Rabia DJELLOULI	Professeur, California State University	Président du jury
Mme Laurence HALPERN	Professeur, Université Paris 13	Rapporteur
M. Serge PETITON	Professeur, Université de Lille	Examineur
M. Xavier VIGOUROUX	Ingénieur de Recherche, Bull	Examineur

Équipe projet Inria Magique-3D, Centre de recherche Inria Bordeaux Sud-Ouest
Laboratoire de Mathématiques et de leurs Applications de Pau, UMR CNRS 5142

Remerciements

Je tiens à remercier en premier lieu et avec insistance H  l  ne BARUCQ, la directrice de cette th  se, pour m'avoir fait confiance tout au long de cette aventure. J'ai beaucoup appr  ci   sa rigueur scientifique, son franc-parler en toute circonstance et aussi son rire tonitruant. Je remercie avec tout autant d'insistance Julien DIAZ, avec qui j'ai pass   tant d'heures sur les formules et sur le code, bien plus que le nombre de bi  res partag  es par la suite. Je reste impressionn   par sa patience    toute   preuve et le large   ventail de ses connaissances. Je remercie   galement Henri CALANDRA, pour la libert   qu'il m'a laiss  e sur le sujet de la th  se et pour la dimension industrielle concr  te qu'il a apport  e. Je peux dire que j'ai eu une sacr  e chance d'  tre encadr   par ces trois personnes, ce qui a rendu ces ann  es de th  se si agr  ables et m  morables.

Je souhaite remercier en suivant George BOSILCA et Emmanuel AGULLO pour leur implication sur la seconde partie de la thèse. George m'a fait profiter de son expertise notoire, sans mettre de hiérarchie entre nous, et Emmanuel a joué le rôle de coordinateur et de relecteur sur le manuscrit. Cette collaboration internationale a été une expérience très enrichissante autant sur le plan personnel que professionnel.

Je remercie maintenant les rapporteurs de cette thèse, Laurence HALPERN et Frédéric DESPREZ, tous deux experts dans leurs domaines spécifiques *a priori* distincts et qui ont accepté de relire ce travail essayant de les fusionner. Merci également aux autres membres du jury, Serge PETITON, Xavier VIGOUROUX et Rabia DJELLOULI, qui a accepté de présider cette équipe éclectique.

J'aimerais aussi remercier toutes les personnes avec qui j'ai échangé durant ces trois années, les ingénieurs de Total notamment Paul WILLIAMSON, Bertrand DENEL, Long QU et Corentin ROSSIGNON, les ingénieurs de SGI, Pierre-Eric BERNARD et Didier REMY, mais également Reda BAINA, Ludovic METIVIER et sûrement d'autres que j'oublie. Nos discussions m'ont permis d'avancer sur la compréhension et sur l'étayage de ce travail.

Merci bien entendu à mes collègues des équipes Inria de PAU, Magique3D et Cagire, surtout Maxime MOGE, ainsi qu'aux doctorants du Laboratoire de Mathématiques Appliquées de l'université avec qui j'ai partagé des repas et des cafés.

Je remercie pour finir ma femme, Stéphanie, qui m'a soutenu dès le départ et encouragé au quotidien, et ma fille, Lou-Anne, qui n'a pas été souvent malade et qui a fait ses nuits rapidement. Je vous aime.

Remerciements

Table des matières

Liste des figures	xi
Liste des tableaux	xiii
Liste des algorithmes	xv
Introduction générale	1
I Conditions aux limites pour les ondes élastiques anisotropes	7
Introduction à la partie I	9
1 Modélisation des équations d'ondes en milieu transverse isotrope	13
1.1 Le système de l'élastodynamique	13
1.1.1 Formulation des équations homogènes	13
1.1.2 Terme source et condition initiale	14
1.2 Anisotropie de type TTI	17
1.2.1 Définition et propriétés physiques	17
1.2.2 Equations de l'élastodynamique dans un milieu anisotrope	22
1.3 Discrétisation des équations	23
1.3.1 Définition et état de l'art	23
1.3.2 Discrétisation en espace	26
1.3.3 Discrétisation en temps	31
1.4 État de l'art sur la simulation d'ondes anisotropes pour des phénomènes géophysiques	32
2 Conditions aux limites absorbantes pour des matériaux VTI	37
2.1 Construction d'une CLA isotrope par diagonalisation	37
2.1.1 Construction d'une CLA pour un matériau isotrope 2D	37
2.1.2 Étude expérimentale pour l'élastique isotrope 2D	42
2.2 Construction d'une CLA VTI par diagonalisation	48
2.2.1 Construction d'une CLA pour un matériau VTI 2D	48
2.2.2 Construction de CLA VTI 2D par découplage des ondes	52
2.2.3 Étude expérimentale pour l'élastique VTI 2D	56
2.3 Stabilité de la CLA VTI 2D	65
2.3.1 Stabilité des problèmes mixtes continu, semi-discret et discret	65
2.3.2 Courbes d'énergies des études isotrope et VTI 2D	70
2.4 Coefficients de réflexion de la CLA VTI 2D	78
2.5 Conclusion	82

3	Conditions aux limites absorbantes pour des matériaux TTI	87
3.1	Processus de construction en 2D	87
3.1.1	Remarques préliminaires	87
3.1.2	Construction d'une CLA pour un matériau TTI 2D	88
3.1.3	Étude expérimentale pour l'élastique TTI 2D	97
3.1.4	Stabilité des problèmes mixtes TTI 2D	106
3.1.5	Courbes d'énergies de l'étude TTI 2D	107
3.2	Coefficients de réflexion de la CLA TTI 2D	112
3.3	Extension au cas 3D	116
3.3.1	Construction d'une CLA pour un matériau TTI 3D	116
3.3.2	Étude expérimentale préliminaire pour l'élastique TTI 3D	129
3.3.3	Stabilité des problèmes mixtes TTI 3D	137
3.3.4	Courbes d'énergies de l'étude préliminaire TTI 3D	138
3.3.5	Cas d'une frontière artificielle plane générale	138
3.4	Conclusion	142
	Conclusion à la partie I	149
II	Programmation à base de tâches sur support d'exécution	153
	Introduction à la partie II	155
	Vocabulaire introductif	159
4	Étude du code DIVA initial pour la propagation des ondes	163
4.1	Implémentation par passage de message	163
4.1.1	Pseudo-code séquentiel à partir de la discrétisation de la partie I . . .	163
4.1.2	Pseudo-code parallèle par passage de messages	164
4.2	Étude de performance	167
5	Programmation en tâches et support d'exécution	169
5.1	Paradigme de parallélisme par tâches	169
5.1.1	Du code séquentiel au graphe de tâches	170
5.1.2	Différents modèles de programmation à base de tâches	170
5.2	État de l'art des supports d'exécution pour la programmation à base de tâches	173
5.2.1	Petit historique des supports d'exécution	173
5.2.2	Classification et utilisation par la communauté	175
5.3	Le support d'exécution PaRSEC	176
6	Formulation à base de tâches du code de propagation des ondes	179
6.1	DAG et description JDF des tâches du code DIVA	179
6.1.1	DAG du code DIVA	179
6.1.2	Description paramétrée en JDF	180
6.2	Extensions pour architectures à mémoire partagée	183
6.2.1	Granularité et localité	183
6.2.2	Illustration expérimentale	184
6.2.3	Gestion des nœuds ccNUMA	186

7 Étude expérimentale	187
7.1 Description des cas tests	187
7.2 Résultats numériques	189
7.2.1 Machines à un nœud ccNUMA	189
7.2.2 Coprocesseur Intel Xeon Phi	192
7.3 Prochaines étapes	194
Conclusion à la partie II	195
A Annexes à la partie I	199
A.1 Rotation des matrices d'élasticité anisotrope	199
A.2 Code Maple TM pour les valeurs propres et vecteurs propres VTI 2D	200
A.3 Code Maple TM pour la simplification à l'ordre 0 d'un système	202
A.4 Code Maple TM pour les coefficients de réflexion VTI 2D	204
A.5 Code Maple TM pour les courbes de lenteur	207
A.6 Code Maple TM pour les coefficients de réflexion TTI 2D	209
A.7 Code Maple TM pour les surfaces de lenteur	213
Bibliographie	217

Table des matières

Table des figures

1	Acquisition des données sismiques, terrestre et maritime	2
2	Corrélation des techniques numériques d'exploration sismique et de la complexité des milieux considérés avec les évolutions de la puissance HPC	3
3	Exemple de RTM acoustique 2D sur Marmousi	4
4	Illustration de la compression et du cisaillement des ondes P et S, source Internet	5
1.1	Ondelette de Ricker pour une fréquence pic de $20Hz$	15
1.2	Schémas de matériaux isotrope, TI et orthotrope	18
1.3	Photographie de couches géologiques TI	19
1.4	Angles d'inclinaison TTI	20
1.5	Exemples de fronts d'ondes élastiques anisotropes	22
1.6	Exemples de fronts d'ondes S singuliers	22
1.7	Fonctions discrètes sur un maillage à une dimension, continues et discontinues	25
1.8	Localisation des degrés de liberté en deux dimensions, pour trois discrétisations	25
1.9	Illustration de l'adaptabilité <i>hp</i> sur un maillage en deux dimension	25
1.10	Passage à l'élément de référence en 2D	30
1.11	Modèles d'anisotropie du cas test synthétique BP	33
1.12	Comparaison isotrope/TTI sur le cas test synthétique BP	33
1.13	Exemple de RTM élastique isotrope 2D	34
2.1	Cas tests isotropes 2D homogène et bi-couches, bande et coin	42
2.2	Module du vecteur vitesse au cours du temps sur la configuration homogène isotrope 2D	45
2.3	Module du vecteur vitesse au cours du temps sur la configuration bi-couche bande isotrope 2D	46
2.4	Module du vecteur vitesse au cours du temps sur la configuration bi-couche coin isotrope 2D	47
2.5	Cas tests VTI 2D homogène, bi-couches et coin	56
2.6	Module du vecteur vitesse au cours du temps sur la configuration homogène VTI 2D	60
2.7	Module du vecteur vitesse au cours du temps sur la configuration bi-couche bande VTI 2D	61
2.8	Module du vecteur vitesse au cours du temps sur la configuration bi-couche coin VTI 2D	62
2.9	Comparaison avec la CLA isotrope sur la configuration homogène VTI 2D . .	63
2.10	Module du vecteur vitesse pour un cas 2D VTI non-elliptique homogène . . .	64
2.11	Comparaison avec la CLA isotrope pour un cas 2D VTI non-elliptique homogène	64

Table des figures

2.12	Énergie discrète pour les configurations isotrope 2D	73
2.13	Énergie discrète pour les configurations VTI 2D	74
2.14	Comparaison avec la CLA isotrope dans un milieu homogène VTI 2D	75
2.15	Énergie discrète dans les cas VTI 2D non-elliptique	76
2.16	Comparaison avec la CLA isotrope sur un cas VTI 2D non-elliptique	76
2.17	Énergie discrète pour des temps long dans les cas isotrope, VTI elliptique et VTI non-elliptique 2D	77
2.18	Schéma des angles intervenant dans la réflexion de l'onde P et de l'onde S, en VTI	79
2.19	Coefficients de réflexion de la CLA VTI 2D et comparaison	82
3.1	Exemples de courbes de lenteur élastiques isotropes, VTI et TTI	90
3.2	Cas tests TTI 2D homogène, bi-couches et coin	97
3.3	Module du vecteur vitesse au cours du temps sur la configuration homogène TTI 2D	101
3.4	Module du vecteur vitesse au cours du temps sur la configuration bi-couche bande TTI 2D	102
3.5	Module du vecteur vitesse au cours du temps sur la configuration bi-couche coin TTI 2D	103
3.6	Comparaison avec la CLA isotrope dans la configuration homogène TTI 2D	104
3.7	Module du vecteur vitesse pour un cas 2D TTI non-elliptique homogène	105
3.8	Comparaison avec la CLA isotrope dans un milieu 2D TTI non-elliptique homogène	105
3.9	Énergies discrètes pour les configurations TTI 2D	108
3.10	Comparaison avec les CLA isotrope et VTI dans un milieu homogène TTI 2D	109
3.11	Énergie discrète dans les cas TTI 2D non-elliptique	110
3.12	Comparaison avec les CLA isotrope et VTI sur un cas TTI 2D non-elliptique	110
3.13	Énergie discrète pour des temps long dans les cas TTI elliptique et TTI non-elliptique 2D	111
3.14	Schéma des angles intervenant dans la réflexion de l'onde P et de l'onde S, en TTI	114
3.15	Domaine d'étude des coefficients de réflexion de l'onde P et de l'onde S en TTI	114
3.16	Coefficients de réflexion de la CLA TTI 2D et comparaison	116
3.17	Exemples de surfaces de lenteur élastiques isotropes	117
3.18	Exemples de surfaces de lenteur élastiques VTI	118
3.19	Exemples de surfaces de lenteur élastiques TTI	119
3.20	Cas test TTI 3D homogène	129
3.21	Module du vecteur vitesse au cours du temps dans la configuration homogène TTI 3D, vue 3D	132
3.22	Module du vecteur vitesse au cours du temps sur la configuration homogène TTI 3D, vue 2D dans le plan (y, z)	133
3.23	Module du vecteur vitesse au cours du temps sur la configuration homogène TTI 3D, vue 2D dans le plan (x, z)	133
3.24	Module du vecteur vitesse au cours du temps sur la configuration homogène TTI 3D, vue 2D dans le plan (x, y)	134
3.25	Comparaison avec la CLA isotrope dans la configuration homogène TTI 3D, vue 3D	134

3.26	Comparaison avec la CLA isotrope dans la configuration homogène TTI 3D, vue 2D dans le plan (y, z)	135
3.27	Comparaison avec la CLA isotrope dans la configuration homogène TTI 3D, vue 2D dans le plan (x, z)	135
3.28	Comparaison avec la CLA isotrope dans la configuration homogène TTI 3D, vue 2D dans le plan (x, y)	136
3.29	Comparaison avec les CLA isotrope et VTI dans un milieu homogène TTI 3D	138
3.30	Évolution des processeurs Intel	156
3.31	Architectures des processeurs	156
3.32	Evolution des supercalculateurs du top500	157
3.33	Architectures CPU, GPU et MIC	158
3.34	Illustration des effets potentiels de la granularité	161
3.35	Illustration de l'équilibrage de charge sur deux CPUs, grain fin et grain grossier	161
4.1	Ensemble \overline{K} des degrés de liberté qui interviennent dans la matrice de rigidité locale à la maille K	165
4.2	Exemple de découpage d'un maillage 2D en trois sous-domaines	166
4.3	Trace d'une exécution du code DIVA	167
4.4	Étude de l'efficacité parallèle du code DIVA	168
5.1	DAG du pseudo-code basique	170
6.1	DAG correspondant à une itération temporelle de l'algorithme DIVA pour un maillage découpé en trois sous-domaines comme dans la Fig. 4.2	180
6.2	Exemple de subdivision dans un maillage 2D	184
6.3	<i>Speedup</i> sur un processeur, effets de la granularité et du vol de tâches	185
6.4	Exemple d'architecture ccNUMA à 4 processeurs	186
7.1	Cas test géologique d'un dôme de sel 3D	188
7.2	<i>Speedup</i> sur 20-core Xeon, pour différentes granularités	189
7.3	Résultats 20-core Xeon, meilleur <i>speedup</i> et efficacité	190
7.4	Détails de résultats sur 64-core Xeon pour la prise en compte des facteurs NUMA	191
7.5	Résultats 64-core Xeon, meilleur <i>speedup</i> et efficacité	191
7.6	Comparaison de traces d'exécution sur un exemple 32 cœurs, MPI et PaRSEC	192
7.7	<i>Speedup</i> sur Xeon Phi, pour différentes granularités	193
7.8	Résultats Xeon Phi, meilleur <i>speedup</i> et efficacité	193

Table des figures

Liste des tableaux

1.1	Correspondance des indices entre tenseur et matrice d'élasticité	17
1.2	Nombre de degrés de liberté en deux et trois dimensions	25
2.1	Caractéristiques physiques des cas-tests isotropes	42
2.2	Valeurs maximales du module du vecteur vitesse pour l'étude isotrope 2D . .	44
2.3	Caractéristiques physiques des cas-tests VTI	56
2.4	Valeurs maximales du module du vecteur vitesse pour l'étude VTI 2D	58
3.1	Caractéristiques physiques des cas-tests TTI 2D	97
3.2	Valeurs maximales du module du vecteur vitesse pour l'étude TTI 2D	100
3.3	Caractéristiques physiques du cas test TTI 3D	129
5.1	Classification des supports d'exécution en fonction du modèle de programmation en tâches	175
5.2	Classification des supports d'exécution en fonction des architectures supportées	176
7.1	Caractéristiques des couches géologiques du cas test	187
7.2	Facteurs NUMA de la machine <i>20-core Xeon</i>	188
7.3	Facteurs NUMA de la machine <i>64-core Xeon</i>	189
7.4	Efficacité relative à l' <i>hyperthreading</i> pour la machine <i>Xeon Phi</i>	193

Liste des Algorithmes

2.1	Méthode de la puissance	71
4.1	Pseudo-code séquentiel de DIVA (discrétisation DG + Leap-Frog)	165
4.2	Pseudo-code parallèle de DIVA par passage de messages	166
5.1	Pseudo-code basique en séquentiel	170
5.2	Pseudo-code basique en tâches, modèle séquentiel	171
5.3	Pseudo-code basique en tâches, modèle paramétré	172
6.1	Description JDF de la tâche <i>Compute_V</i>	182
6.2	Description JDF de la tâche <i>Pack_V</i>	182
6.3	Description JDF de la tâche <i>Unpack_V</i>	182
6.4	Description JDF de la tâche <i>Compute_S</i>	183
6.5	Description JDF de la tâche <i>Pack_S</i>	183
6.6	Description JDF de la tâche <i>Unpack_S</i>	183

Introduction générale

Cette thèse de doctorat s’inscrit au sein de l’action stratégique DIP (*Depth Imaging Partnership*)¹ qui est un partenariat entre Inria et Total (<http://dip.inria.fr>) dédié à la recherche long-terme sur le thème de l’imagerie sismique. L’imagerie de la Terre par équations d’ondes est un domaine applicatif qui met en jeu des compétences variées visant à la conception de logiciels performants capables de délivrer des images de qualité des zones explorées. L’exploration pétrolière a commencé il y a bien longtemps et il n’est pas étonnant qu’aujourd’hui les domaines restant à explorer soient fortement hétérogènes, souvent difficiles d’accès. La conception de logiciels performants est donc un outil clé pour l’exploration pétrolière, non seulement pour l’aide à la détection mais aussi pour alerter sur de possibles erreurs d’appréciation des quantités d’hydrocarbures présents dans la zone ciblée. La tâche des logiciels est aujourd’hui particulièrement difficile car les phénomènes de propagation d’ondes sont d’autant plus complexes que les couches géologiques sont fortement contrastées. L’objectif de DIP, et donc en particulier de cette thèse, est d’améliorer les techniques d’imagerie sismique utilisées pour la prospection pétrolière en favorisant les collaborations entre équipes de recherche aux compétences complémentaires.

La recherche de nouvelles ressources d’hydrocarbures est un processus long et complexe. Une étude typique s’étale sur plusieurs mois et nécessite l’intervention de différentes équipes scientifiques : géologues, géophysiciens et géonumériciens. L’étape la plus critique est celle qui fournit une cartographie aussi précise que possible du sous-sol. Elle se base généralement sur la méthode dite de “sismique par réflexion” qui étudie la propagation des ondes sismiques dans le sol et leurs réflexions aux interfaces des différentes couches géologiques.

Une fois qu’un terrain a été ciblé et que des études préliminaires ont été effectuées, la prospection sismique commence par l’étape d’acquisition de données. Un ensemble de sources explosives provoque successivement la propagation d’ondes dans le sol et un quadrillage de récepteurs mesure, en surface, les ondes directes et réfléchies. Ce processus est illustré sur la Fig. 1.

Plusieurs techniques numériques permettent d’obtenir une cartographie du sous-sol en fonction du degré de précision souhaité. Elles se différencient principalement par les coûts de calculs générés, traduisant un réel besoin en ressource informatique pour s’exécuter en un temps raisonnable. La Fig. 2 représente la corrélation entre ces techniques d’imagerie sismique

1. DIP est une action qui a été créée en 2008 à partir de la collaboration entre l’équipe Inria *Magique-3D* (Pau) dirigée par Hélène BARUCQ et Henri CALANDRA de Total EP (Pau/Houston). Deux autres équipes Inria : *Hiepac* (Bordeaux) dirigée par Luc GIRAUD et *Nachos* (Nice) dirigée par Stéphane LANTERI ont depuis rejoint le partenariat.

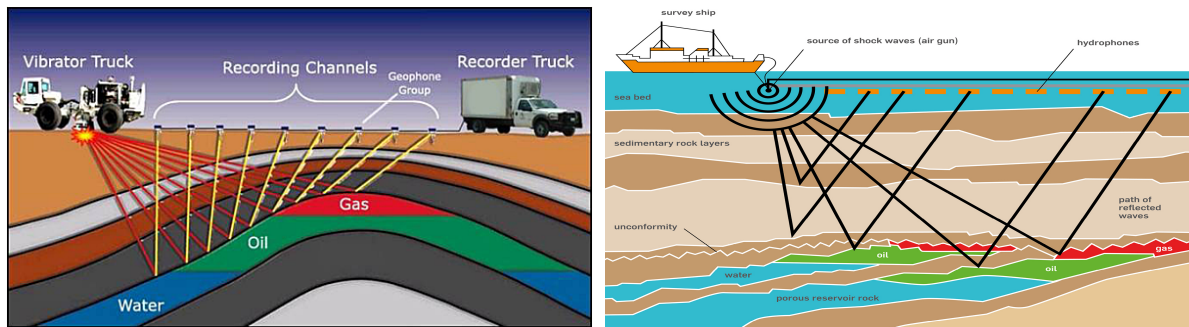


FIGURE 1 – Acquisition des données sismiques, terrestre et maritime, source Internet

et l'évolution des machines de calcul parallèle intensif – ou calcul massivement parallèle. L'imagerie sismique est donc un domaine de prédilection pour ce que l'on appelle HPC (*High-Performance Computing*). La puissance des machines, exprimée en Flop/s (*FL*oating-*point* *O*perations *P*er *S*econd), est représentée sur les courbes, la plus puissante à gauche et celle de Total à droite. L'abscisse s'étale de l'avènement de l'informatique à grande échelle à la fin des années 1980, jusqu'au prochain palier que la communauté espère atteindre en 2020 : l'exaFlop/s (10^{18} Flop/s).

À gauche sont listées les principales méthodes d'exploration sismique utilisées par l'industrie pétrolière. Chacune d'elles s'étale sur une échelle de temps et de puissance, représentée par un rectangle coloré et fait face, à droite, à une échelle de complexité des milieux que l'on peut simuler. La corrélation est, nous semble-t-il, évidente entre la précision recherchée et le coût de calculs. Il en ressort une classification selon des critères de capacité et besoins.

Nous nous situons aujourd'hui dans le rectangle des techniques "*Full Wave Equation*" dont la plus utilisée est la RTM (*Reverse Time Migration*). Cependant, des techniques plus anciennes comme la migration profondeur de Kirchhoff ou l'approximation de Gauss par exemple, peuvent encore être utilisées dans certains cas simplifiés. Quant à l'avenir, c'est vraisemblablement la technique FWI (*Full Waveform Inversion*), reposant sur la résolution du problème inverse, qui devrait émerger. Celle-ci est plus coûteuse mais permet d'être encore plus précis. En effet, la RTM fournit des informations sur la cinétique du milieu tandis que résoudre le problème inverse permet d'obtenir des informations sur la dynamique du milieu, donnant ainsi des informations directes sur ses constituants.

La RTM, communément attribuée à [BKS83, McM83, Whi83], est une des méthodes de migration par extrapolation de l'équation des ondes. Sa spécificité est de considérer l'équation des ondes complète, appelée en anglais "*full wave equation*". Cela lui permet d'être assez précise, même lorsque le sous-sol présente des hétérogénéités géologiques, mais cela la rend aussi très coûteuse en temps et donc en ressource informatique. Les méthodes de migration se basent sur un modèle de vitesse, c'est-à-dire sur une description des différentes couches du sous-sol discriminées uniquement par la vitesse de propagation des ondes les parcourant.

La RTM est initialisée en utilisant un modèle de vitesse approximatif puis deux simulations numériques de propagation des ondes sont effectuées par source. La première reproduit l'expérience de l'acquisition, avec une source numérique imitant le type d'explosion faite sur le terrain. La seconde réalise la propagation d'ondes inverse dans laquelle les données sismiques mesurées sur le terrain servent de source numérique. Appelée rétro-propagation, cette

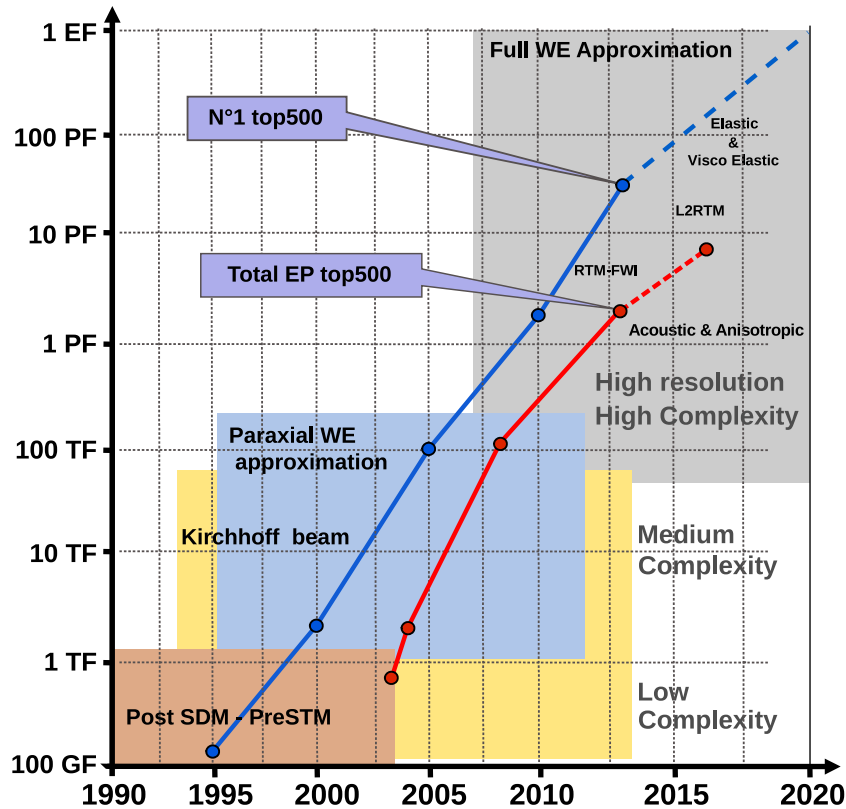


FIGURE 2 – Corrélation des techniques numériques d’exploration sismique et de la complexité des milieux considérés avec les évolutions de la puissance HPC, image reproduite à partir d’un graphique de Calandra, Morton et Etgen en 2011 (référéncé plus tard dans [Wen14])

simulation est justifiée par la réversibilité en temps de l’équation des ondes.

Nous nous servons ensuite du principe d’imagerie, énoncé par Claerbout en 1971 dans [Cla71] et résumée par la phrase : “*reflectors exist at points in the ground where the first arrival of the downgoing wave is time coincident with an upgoing wave*”. Ainsi, en chaque point du milieu où il y a corrélation entre le champ propagé et le champ rétro-propagé, nous pouvons en déduire la présence d’un réflecteur. L’ensemble de ces réflecteurs trace une carte du sous-sol indiquant les différentes interfaces géologiques.

La RTM consiste donc à projeter dans le domaine spatial des données temporelles enregistrées par les récepteurs. La Fig. 3 est un exemple de résultat RTM sur le cas synthétique Marmousi 2D, dont le modèle de vitesse est acoustique.

Une fois adopté le principe selon lequel appliquer la RTM signifie résoudre des équations d’ondes complètes, nous pouvons différencier les techniques de RTM par la méthode numérique utilisée pour résoudre l’équation des ondes. Cependant quelle qu’elle soit, cette technique transforme l’équation physique en un système matriciel du type $A\mathbf{x} = \mathbf{b}$, où A est une matrice inversible, \mathbf{x} est le vecteur des inconnues à déterminer et \mathbf{b} est un vecteur fixé par les données du problème.

Le plus souvent, les industriels pétroliers utilisent dans leur “code en version production”, la méthode des différences finies. Cette discrétisation est structurée, elle permet un parcours

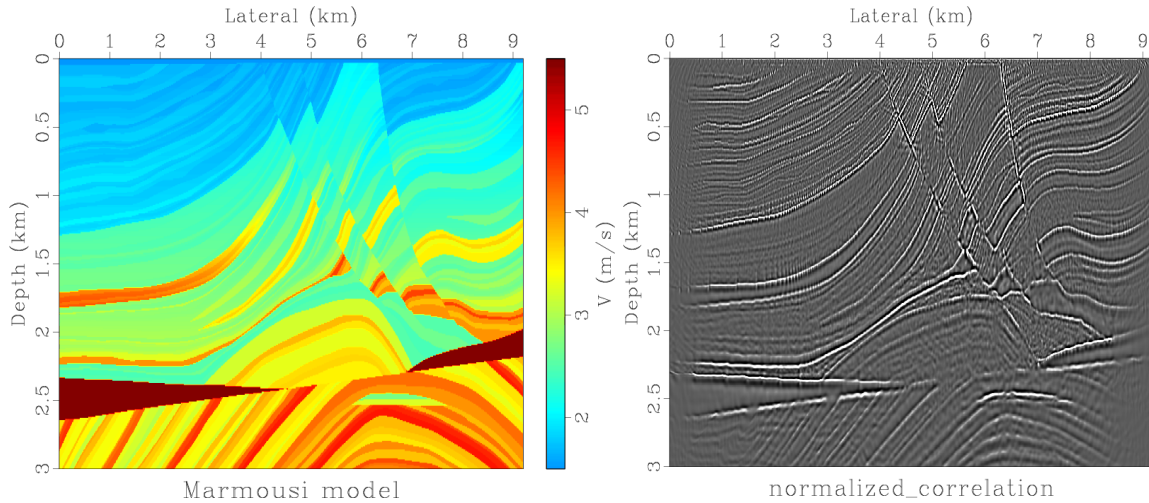


FIGURE 3 – Exemple de RTM acoustique 2D sur Marmousi, modèle de vitesse (gauche), cartographie des réflecteurs (droite) – images issues de [YGW14]

ordonné et prédéfini des données et donc un gain de temps certain lors des calculs. Cependant, la matrice A est dans ce cas multi-diagonale, ce qui complique son inversion notamment lorsque l'ordre de discrétisation est très élevé ou lorsque la taille du problème est très grande. Or, c'est un cas de figure de plus en plus fréquent car les gisements de pétrole sont dans des zones très complexes comme par exemple en eau profonde avec marges abruptes. Cet inconvénient est la particularité des méthodes à *stencil* proportionnel à l'ordre de discrétisation, augmentant le nombre de diagonales de la matrice et nécessitant plus de communications de données pour le calcul parallèle.

Dans le contexte du partenariat DIP, la compagnie pétrolière Total a choisi de privilégier la méthode de Galerkin discontinue dans un code pour l'instant "en version recherche". La discrétisation par DGM (*Discontinuous Galerkin Method*) est non-structurée ce qui facilite grandement le traitement des hétérogénéités mais rend aussi le parcours des données plus aléatoire. Un autre point à l'avantage de cette méthode est qu'elle fournit une matrice A diagonale par blocs, chacun de taille proportionnelle à l'ordre de discrétisation, ce qui permet de l'inverser très facilement. C'est d'ailleurs dans un contexte de calcul parallèle que la DGM prend tout son sens. En effet, elle nécessite, certes, un peu plus de données que la méthode des différences finies, mais les calculs deviennent très indépendants. Ainsi, les problèmes de grandes tailles ne sont plus délicats à traiter, sitôt que la capacité des ressources informatiques est suffisante. Ce sont des méthodes à *stencil* fixe, seulement les données des voisins directs (par les faces) sont nécessaires, quel que soit l'ordre de discrétisation.

Par ailleurs, il existe différents degrés de précision pour la modélisation physique du sous-sol. La version la plus simplifiée considère des matériaux acoustiques isotropes, c'est-à-dire ne dépendant que d'une seule vitesse de propagation. Une version plus réaliste consiste à décrire les matériaux comme élastiques isotropes. Dans ce cas, deux vitesses de propagation entrent en considération : celle des ondes P, dites ondes de compression et celle des ondes S, dites ondes de cisaillement, décrites sur la Fig. 4. Enfin, pour aller plus loin, il est possible de prendre en compte l'anisotropie, de modéliser la dispersion, d'ajouter des fissures et même de considérer les matériaux comme des milieux poreux. Tout cela ajoute bien évidemment de la complexité

à tous les niveaux de la simulation : modélisation, schéma numérique, implémentation, calcul parallèle.

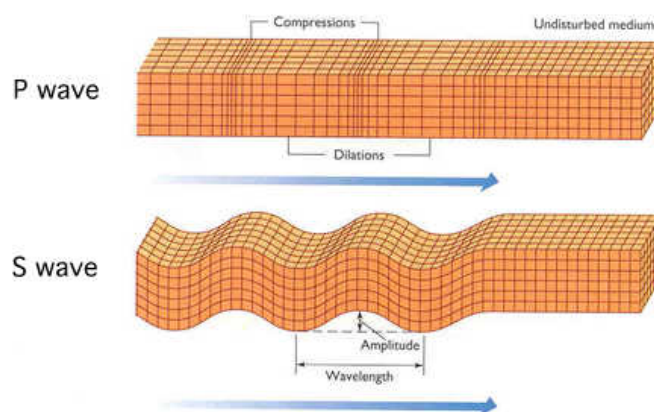


FIGURE 4 – Illustration de la compression et du cisaillement des ondes P et S, source Internet

Le code de “sismique par réflexion” de Total, basé sur la RTM, s’appelle DIVA (*Depth Imaging Velocity Analysis*). Il traite les problèmes 2D et 3D, acoustique et élastique et se sert d’outils internes à Total pour cartographier les réflecteurs. DIVA est codé en Fortran et a été développé pour le paradigme de parallélisme par passage de messages, à travers l’utilisation de la bibliothèque MPI (*Message Passing Interface*). De plus, le code a été fortement optimisé pour tirer parti des performances matérielles des supercalculateurs de Total, basé sur la technologie Intel. Dans sa version production, le code en devient même difficilement lisible. Au sein du partenariat DIP, nous travaillons exclusivement sur la version recherche de DIVA, qui est optimisée HPC également mais qui doit rester facilement accessible aux chercheurs.

Cette thèse a pour objectif de contribuer à améliorer l’imagerie par équations d’ondes élastiques dans des milieux anisotropes. Il s’agit d’un sujet de recherche très vaste et nous avons choisi de cibler deux contributions possibles qui, si elles supposent des compétences très différentes, ont la même finalité : réduire les coûts de calcul requis pour simuler la propagation d’ondes élastiques en milieu anisotrope. Les deux parties peuvent être lues indépendamment.

Dans une première partie, la simulation est rendue plus réaliste en modélisant le sous-sol comme un matériau hétérogène élastique anisotrope. L’anisotropie est de type TTI (*Tilted Transverse Isotropic*), communément utilisée en géophysique. Le modèle physique n’est donc pas nouveau mais sa mise en œuvre introduit la problématique des conditions aux limites à appliquer au système. Les solutions existantes pour les matériaux isotropes ne sont pas directement transposables, notamment les PML (Perfectly Matched Layer), sur lesquelles il a été démontré qu’elles génèrent des instabilités numériques dans ce cas. De plus, le contexte HPC ajoute des contraintes supplémentaires car il faut veiller à ne pas détruire les performances des calculs en utilisant des conditions aux limites dont l’efficacité ne serait garantie qu’avec un nombre important de communications. L’utilisation de PML ou de conditions d’ordre élevé peut notamment détruire considérablement les performances des algorithmes parallèles. Cette partie décrit la construction d’une condition aux limites absorbante pour les ondes élastiques TTI 3D et sa validation numérique dans le code de Total. Notre objectif est de fournir à la

communauté géophysique une condition aux limites stable et efficace dans un contexte HPC. C'est pourquoi nous nous focalisons sur une condition d'ordre peu élevé.

Dans une deuxième partie, le paradigme de parallélisme du code de propagation d'ondes élastiques est optimisé pour la portabilité. L'évolution des architectures matérielles amène une hétérogénéité des ressources informatiques et donc une complexité croissante pour les développeurs souhaitant atteindre les performances optimales des machines HPC. Le paradigme de parallélisme par tâches est une solution qui a fait ses preuves dans les bibliothèques d'algèbre linéaire notamment. Les codes parallèles par tâches sont plus flexibles, portables et sont généralement ordonnancés par un support d'exécution, en anglais *runtime*. Cependant, le portage d'un code industriel, avec ses contraintes propres, n'avait pas encore été réalisé jusqu'à présent. Cette partie² décrit la réécriture de l'algorithme du code de Total en tâches et étudie le comportement sur différentes architectures machines. Notre objectif est de montrer la faisabilité d'un portage vers une programmation à base de tâches pour un code industriel.

Les améliorations proposées, bien que totalement indépendantes, se révèlent au final très complémentaires. En effet, l'introduction de l'anisotropie pour rendre la simulation plus réaliste augmente le nombre des calculs de façon non-uniforme, ce qui détériore un peu plus leur répartition sur les ressources informatique. L'utilisation de la programmation à base de tâches, au-delà d'offrir de la flexibilité et de la portabilité, permet de corriger en partie ce déséquilibre et donc de réduire le temps nécessaire à la simulation.

Les expériences numériques ont été réalisées sur plusieurs calculateurs, la machine expérimentale co-développée par Inria PlaFRIM (Plateforme Fédérative pour la Recherche en Informatique et Mathématiques - <https://plafrim.bordeaux.inria.fr>), la machine de calculs de l'université de Bordeaux MCIA (Mésocentre de Calcul Intensif Aquitain - www.mcia.univ-bordeaux.fr) et pour des besoins spécifiques, le supercalculateur de Total Pangea (www.top500.org/system/178071).

2. Cette deuxième partie a été réalisée dans le cadre d'une collaboration internationale avec George BOSILCA de l'ICL (*Innovative Computing Laboratory*), Knoxville-Tennessee USA, sous la coordination d'Emmanuel AGULLO de l'Inria, équipe *Hiepacs*, grâce à l'action stratégique DIP.

Première partie

Conditions aux limites pour les ondes élastiques anisotropes

Introduction à la partie I

La propagation des ondes élastiques est un phénomène physique décrit par des équations linéaires. Les sources explosives utilisées en “sismique par réflexion” sont calibrées pour l’étude d’une petite partie de la croûte terrestre, de l’ordre d’une dizaine de kilomètres de profondeur sur quelques kilomètres carrés de surface. Les longueurs d’ondes sont donc très petites, de l’ordre de plusieurs dizaines de mètres³. Pourtant, les ondes se propagent au-delà de ce domaine physique, mais avec des intensités affaiblies par dissipation. Cela n’affecte donc que marginalement les mesures en surface.

Numériquement, le domaine discret est un maillage de points en trois dimensions, reliés en cellules. La longueur d’onde du problème physique définit directement la taille maximale de ces cellules, conditionnant la densité du maillage. En effet, pour pouvoir simuler numériquement une onde, il est d’usage d’avoir au moins une dizaine de points par longueur d’onde. Le rapport entre les dimensions du domaine physique et la longueur d’onde des sources explosives se traduit donc par un maillage très dense. Aussi, le domaine discret est réduit à son minimum, décrivant un pavé droit dont les dimensions sont celles du domaine d’étude physique.

La frontière externe du domaine discret n’existe pas dans le domaine physique qui est un milieu ouvert. Pour les simulations, les équations de départ doivent être couplées à des conditions aux limites agissant sur cette frontière artificielle. Le problème mixte obtenu doit être bien posé, c’est-à-dire admettre une solution unique dépendant des données⁴. Cette solution doit être aussi proche que possible de la restriction de la solution du problème initial.

Une condition aux limites est appelée “transparente” lorsque la solution du problème mixte correspondant est exactement la restriction de la solution du problème initial dans le domaine de calcul. Cette condition fait généralement intervenir des opérateurs pseudo-différentiels non-locaux, en espace et en temps. Numériquement, cela signifie que la condition agissant sur la frontière dépend d’autres points à l’intérieur du maillage, au pas de temps courant, mais aussi à tous les pas de temps précédents. Outre des difficultés de mise en œuvre souvent causées par une caractérisation abstraite des opérateurs à partir de leur symbole, une condition aux limites transparente génère donc des coûts de calcul souvent rédhibitoires.

En pratique, la condition aux limites transparente est remplacée par une approximation que l’on appelle alors Condition aux Limites Absorbante ou CLA ou ABC (*Absorbing Boundary Condition*) en anglais. La CLA est souvent associée à son ordre qui correspond à l’ordre d’approximation appliqué à la condition transparente. Sans surprise, quand elle est stable, une CLA d’ordre élevé génère moins de réflexions qu’une CLA d’ordre faible.

3. Les ondes émises par les sources explosives ont une vitesse V de quelques milliers de mètres par seconde et une fréquence f de quelques dizaines de Hertz, ce qui fixe l’ordre de grandeur de leur longueur d’onde $l = \frac{V}{f}$.

4. Selon la définition commune de Hadamard pour les systèmes d’équations aux dérivées partielles.

Une bibliographie exhaustive des CLA serait trop longue à développer ici notamment parce que le sujet est traité depuis de nombreuses années et la littérature sur le sujet est vaste. Citons tout de même quelques auteurs qui nous semblent importants pour nos travaux. Les articles de références sont, à notre avis, ceux d’Engquist et Majda [EM77, EM79] datant de la fin des années 1970, où est proposée une technique de construction de CLA d’ordre élevé pour les ondes acoustiques. Ils ont été suivi par les travaux de Reynolds [Rey78] et Halpern [Hal80] qui proposent des CLA d’ordre 1 pour les ondes élastiques. Ce problème est plus difficile à considérer car il s’agit d’un problème vectoriel dans lequel deux types d’ondes interagissent. S’ensuivent beaucoup de travaux apportant améliorations et extensions à d’autres systèmes, notamment ceux d’Higdon [Hig91] sur des conditions d’ordre élevé pour les ondes élastiques et ceux de Collino [Col93] pour les ondes acoustiques. Puis, au début des années 2000, les travaux de Tsokga [Tso99], Hagstrom [Hag03] et Givoli [Giv03] ont permis d’améliorer les simulations numériques dans des milieux isotropes.

Toutes ces références concernent des milieux de propagation isotropes. L’objectif de cette partie I est de rendre la simulation plus réaliste en modélisant le sous-sol comme un matériau hétérogène élastique anisotrope. À notre connaissance, aucune CLA n’a jamais été formulée pour répondre à ce problème.

Une autre voie que celle des CLA consiste à étendre le domaine discret en y ajoutant une couche dans laquelle les ondes seraient atténuées. C’est le principe des couches absorbantes qui est souvent privilégié dans le milieu industriel. Une implémentation simple, que l’on appelle *taper*, peut facilement être utilisée mais génère des réflexions parasites à l’interface entre le domaine d’étude discret et la couche absorbante. Au début des années 1990, Bérenger aborde ce problème pour les ondes électromagnétiques dans [Ber94, Ber96] et propose un moyen de ne générer aucune réflexion à l’interface. C’est l’avènement des couches parfaitement adaptées, en anglais PML (*Perfectly Matched Layers*), qui révolutionnent la simulation numérique de la propagation des ondes.

Cependant, l’addition d’une couche à un maillage ne se fait pas sans contrepartie, surtout en trois dimensions. Cela entraîne un surcoût évident en terme de calculs, mais aussi de stockage, ce qui devient très problématique pour des données de grande taille. De plus, une implémentation en parallèle nécessite des communications spécifiques additionnelles. Sur ce point précis du parallélisme, l’utilisation de CLA n’est pas non plus sans contraintes si les conditions sont d’ordre élevé. Toutefois, lorsque des conditions d’ordre peu élevé suffisent, les communications utiles aux CLA se limitent aux nœuds de la frontière artificielle, ce qui les rend très compétitives par rapport aux couches absorbantes.

Au-delà de ces considérations pratiques, les PML souffrent d’un problème de stabilité dans certains milieux anisotropes (voir [BFJ03]), se traduisant comme une croissance exponentielle de la solution numérique en temps long. En particulier les PML sont instables pour la plupart des milieux TTI (*Tilted Transverse Isotropy*). Ce type de milieu est à considérer pour des applications concrètes car il inclut l’anisotropie des couches du sous-sol.

L’objectif de cette première partie consiste donc à construire une CLA pour les ondes élastiques dans un milieu anisotrope. Pour cela, nous nous imposons deux contraintes. Premièrement, la CLA doit être compatible avec le contexte RTM dans lequel la simulation intervient, c’est-à-dire assurer une stabilité numérique en temps longs. Deuxièmement, la CLA doit facilement s’intégrer dans le code parallèle DIVA, basé sur la méthode de Galerkin

Discontinue DGM.

Comme nous l'avons déjà dit, les CLA d'ordre élevé pour les ondes élastiques, même isotropes, sont difficiles à formuler et à implémenter, surtout dans un contexte de calcul parallèle intensif - HPC. Les CLA d'ordre faible ne souffrent pas de cet inconvénient mais, bien évidemment, c'est au prix des réflexions parasites générées par la frontière artificielle dans le domaine d'étude. Cependant, il ne faut pas oublier que notre objectif est de développer un propagateur optimisé dédié à la RTM qui comporte une étape de sommation capable de filtrer les valeurs incohérentes. L'impact des réflexions est donc minimisé, ce qui donne un intérêt certain pour des CLA d'ordre peu élevé.

Nous allons donc consacrer cette partie du manuscrit à la construction d'une CLA 2D et 3D pour un milieu anisotrope TTI. Cette CLA sera d'ordre peu élevé et stable en temps long. Cette étude se divise en cinq chapitres.

Le chapitre 1 introduit le système de l'élastodynamique pour les milieux anisotropes. La section 1.2 définit l'anisotropie de type TTI et la section 1.3 détaille la discrétisation par la méthode de Galerkin Discontinue DGM. Le chapitre 2 revient sur la construction d'une CLA d'ordre faible pour des milieux anisotropes simplifiés de type VTI (*Vertical Transverse Isotropy*). La méthodologie classique, basée sur celle d'Engquist et Majda, est rappelée à la section 2.1 pour les milieux isotropes et appliquée dans des milieux VTI à la section 2.2.

Le chapitre 3 est consacré à la construction d'une nouvelle CLA d'ordre faible, adaptée pour les milieux TTI. La section 3.1 décrit le processus dans le cas 2D. La stabilité de cette nouvelle CLA est ensuite étudiée section 3.1.4. Nous montrons ensuite qu'on peut étendre notre approche au cas 3D qui fait l'objet de la section 3.3. Des tests numériques, détaillés en 2D et préliminaires en 3D illustrent les bonnes performances des nouvelles CLA.

Chapitre 1

Modélisation des équations d'ondes en milieu transverse isotrope

Ce chapitre introduit le contexte physique et mathématique dans lequel nous évoluerons dans la suite du manuscrit. Après avoir rappelé les équations générales gouvernant la propagation des ondes élastiques, nous détaillons les caractéristiques des milieux élastiques anisotropes, notamment transverse isotrope (TTI), ainsi que leur impact sur les équations. Nous décrivons ensuite la technique de discrétisation utilisée, qui combine une méthode de Galerkin discontinue (DGM) en espace et un schéma Leap-Frog en temps. Ce choix mène à un système linéaire dont la structure des matrices est précisée, justifiant au passage son adéquation avec le cadre de résolution parallèle intensif HPC. Nous terminons ce chapitre par un état de l'art des méthodes géophysiques employées pour prendre en compte l'anisotropie du sous-sol. Le plus souvent partielles ou complexes à mettre en œuvre, aucune d'entre-elles ne propose des conditions de bord absorbant adaptées à la simulation des ondes anisotropes.

1.1 Le système de l'élastodynamique

1.1.1 Formulation des équations homogènes

Nous supposons que le domaine d'étude est un carré (pavé) droit, noté Ω , en dimension deux (trois). La variable d'espace est notée $\mathbf{x} \in \Omega$ et la variable temporelle est $t \in [0, T]$, où $T > 0$ désigne la durée de simulation.

Le système de l'élastodynamique est formé de trois équations qui peuvent être formulées en vitesse ou en déplacement. Nous avons fait le choix de travailler avec le système basé sur la variable vitesse qui est notée \mathbf{v} . Soient ρ la masse volumique des matériaux et $\underline{\underline{\sigma}}$ le tenseur des contraintes. La première équation du système est celle du mouvement qui se déduit de la loi de Newton. Sa forme homogène, c'est-à-dire sans terme source, s'écrit :

$$\rho(\mathbf{x})\partial_t \mathbf{v}(\mathbf{x}, t) = \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \quad (1.1)$$

où “ $\nabla \cdot$ ” désigne l'opérateur divergence. La deuxième équation décrit la déformation des matériaux. Elle fait donc intervenir le tenseur des déformations noté $\underline{\underline{\epsilon}}$. Sous l'hypothèse dite de petites déformations, ce tenseur s'écrit :

$$\underline{\underline{\epsilon}}(\mathbf{v}) = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \quad (1.2)$$

où “ ∇ ” désigne l’opérateur gradient. Pour finir, la troisième équation, appelée loi de Hooke, tient compte des contraintes des matériaux en exprimant les relations du tenseur des contraintes $\underline{\underline{C}}$ (appelé aussi tenseur d’élasticité) avec le tenseur des déformations. Elle s’écrit en formulation vitesse :

$$\partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) = \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) \quad (1.3)$$

où le produit entre le tenseur d’ordre 4 et le tenseur d’ordre 2 est défini par $\partial_t \sigma_{ij} = \sum_{k,l} C_{ijkl} \epsilon_{kl}$.

Le système de l’élastodynamique s’obtient alors en injectant les équations (1.2) et (1.3) dans la dérivée en temps de (1.1) :

$$\rho(\mathbf{x}) \partial_t^2 \mathbf{v}(\mathbf{x}, t) = \nabla \cdot (\underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t))) \quad (1.4)$$

Il s’agit d’une formulation d’ordre deux car elle fait intervenir des dérivées secondes sur la variable vitesse. Il est aussi possible d’écrire le système au premier ordre en considérant le tenseur des contraintes comme variable auxiliaire, ne faisant ainsi intervenir que des dérivées premières. Les inconnues sont alors le vecteur vitesse \mathbf{v} et le tenseur des contraintes $\underline{\underline{\sigma}}$. Une formulation en vitesse-contrainte de l’élastodynamique s’écrit ainsi :

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) \end{cases} \quad (1.5)$$

Dans le projet DIP, la formulation du premier ordre (1.5) a été privilégiée car elle présente l’avantage de donner un accès direct aux quantités utiles pour l’imagerie. Toutefois, l’introduction des équations d’ordre deux est justifiée car elles peuvent être utilisées pour éliminer le tenseur des contraintes qui n’est qu’une variable auxiliaire pour le phénomène à décrire.

1.1.2 Terme source et condition initiale

Les ondes sont excitées par une source ponctuelle modélisée en général comme le produit d’une fonction temporelle et d’une fonction spatiale. La source apparaît physiquement dans l’équation du mouvement (1.1) et se retrouve donc directement dans la formulation du premier ordre (1.5) sur la première équation :

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) + \tilde{s}(t) \mathbf{s}(\mathbf{x}) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) \end{cases} \quad (1.6)$$

où \tilde{s} est une fonction scalaire temporelle et \mathbf{s} est une fonction vectorielle spatiale. Ce système peut se réécrire de façon équivalente avec la source dans la deuxième équation :

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) + s(t) \underline{\underline{s}}(\mathbf{x}) \end{cases} \quad (1.7)$$

où s est une fonction scalaire temporelle (précisément telle que $s = \partial_t \tilde{s}$) et $\underline{\underline{s}}$ est une fonction tensorielle spatiale (précisément telle que $\nabla \cdot \underline{\underline{s}} = \mathbf{s}$).

Code DIVA

Le code DIVA se base sur le choix arbitraire (1.7), c'est-à-dire avec la source sur l'équation du tenseur des contraintes. Dans sa version initiale (i.e. isotrope), la source ne génère que des ondes P. Cela se traduit par une action sur les termes diagonaux du tenseur et la fonction spatiale \underline{s} de la source utilisée s'écrit alors :

$$\underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (1.8)$$

où \mathbf{x}_0 est le point d'impact de la source et $\delta_{\mathbf{x}_0}$ est la distribution de Dirac en ce point. Cependant, dans des milieux anisotropes, cette source génère une onde S de faible intensité en plus d'une onde P.

Afin de reproduire numériquement chaque explosion effectuée sur le terrain, la fonction temporelle s de la source décrit une impulsion sismique communément représentée par une ondelette de Ricker [Ric53] s_R , dérivée seconde d'une fonction gaussienne, voir Fig. 1.1. Elle dépend d'une fréquence pic f_p et son amplitude varie suivant la formule :

$$s_R(t) = (1 - 2\pi^2 f_p^2 t^2) e^{-\pi^2 f_p^2 t^2} \quad (1.9)$$

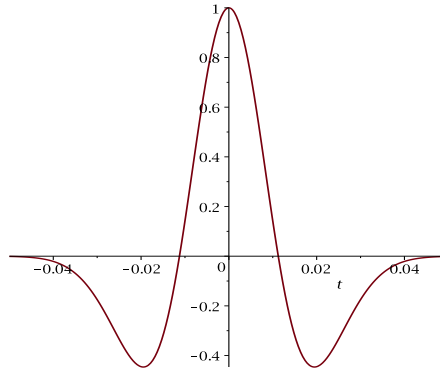


FIGURE 1.1 – Ondelette de Ricker pour une fréquence pic de $20Hz$

Spécifications

Dans le cadre de notre travail sur les CLA, il est intéressant de ne générer qu'une seule des ondes, P ou S, pour étudier plus facilement les réflexions parasites. De plus, comme dans le code DIVA, les géophysiciens utilisent généralement des modèles pseudo-acoustiques, c'est-à-dire se focalisant sur une seule des deux ondes dans un milieu élastique. Ce point sera développé par la suite. Nous allons donc considérer trois configurations d'impulsion sismique : une onde P, une onde S et la combinaison d'une onde P et d'une onde S. Ces cas seront surnommés "pseudo-acoustiques" pour les deux premiers et "élastique" pour le dernier.

Par ailleurs, la propagation des ondes P et S dans un milieu élastique peut se faire à l'aide d'une source ponctuelle ou d'une condition initiale. Dans ce dernier cas, le système du premier

ordre (1.5) s'écrit :

$$\begin{cases} \rho(\mathbf{x})\partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) \\ \mathbf{v}(0, \mathbf{x}) &= \mathbf{v}_0(\mathbf{x}) \\ \underline{\underline{\sigma}}(0, \mathbf{x}) &= \underline{\underline{\sigma}}_0(\mathbf{x}) \end{cases} \quad (1.10)$$

et il s'agit alors de définir \mathbf{v}_0 et $\underline{\underline{\sigma}}_0$.

Nous partons du fait suivant. Toute solution de l'équation des ondes élastiques s'écrit sous la forme :

$$\vec{U} = \vec{\phi} + \vec{\psi} \quad (1.11)$$

où $\vec{\phi}$ correspond à une onde P et $\vec{\psi}$ correspond à une onde S (simple en 2D ou double en 3D).

L'étude de la matrice associée à l'équation de dispersion du système (1.4) admet deux valeurs propres λ_p et λ_s correspondant à l'onde P et à l'onde S. Les vecteurs propres associés peuvent s'exprimer à l'aide d'opérateurs en espace B_p et B_s comme suit :

$$\vec{U} = B_p(g(\mathbf{x})) + B_s(g(\mathbf{x})) \quad (1.12)$$

où $g(\mathbf{x})$ est une distribution quelconque. Il devient alors possible de ne générer qu'une onde P ou une onde S en choisissant par exemple comme condition initiale $\mathbf{v}_0 = B_\star(g)$ et en fixant $\underline{\underline{\sigma}}_0 = 0$. Nous choisissons $g(\mathbf{x}) = e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}$, une fonction gaussienne, qui est une condition initiale classique en propagation d'ondes. Les conditions initiales

$$\mathbf{v}_0(\mathbf{x}) = B_\star(e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}) \quad (1.13)$$

seront détaillées pour chaque configuration isotrope et anisotrope.

Si nous souhaitons maintenant utiliser une source ponctuelle comme dans le système (1.7), il est possible de la construire à partir de la condition initiale. Commençons par observer que le système à l'ordre deux ne générera qu'une onde P ou une onde S en appliquant à une source quelconque $f(\mathbf{x}, t)$ l'opérateur B_p ou B_s respectivement :

$$\rho(\mathbf{x})\partial_t^2 \mathbf{v}(\mathbf{x}, t) = \nabla \cdot (\underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t))) + B_\star(f(\mathbf{x}, t)) \quad (1.14)$$

La reformulation à l'ordre deux du système (1.7) donne :

$$\rho(\mathbf{x})\partial_t^2 \mathbf{v}(\mathbf{x}, t) = \nabla \cdot (\underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t))) + s_R(t) \nabla \cdot \underline{\underline{s}}(\mathbf{x}) \quad (1.15)$$

En choisissant par exemple $f(\mathbf{x}, t) = s_R(t) \delta_{\mathbf{x}_0}(\mathbf{x})$, il reste à définir $\underline{\underline{s}}$ par identification terme à terme suivant l'égalité :

$$\nabla \cdot \underline{\underline{s}}(\mathbf{x}) = B_\star(\delta_{\mathbf{x}_0}(\mathbf{x})) \quad (1.16)$$

Ainsi, pour chaque condition initiale formulée à l'aide des opérateurs B_\star (provenant de l'étude des vecteurs propres de la matrice de dispersion) pour le système (1.10), nous déterminerons une source ponctuelle pour le système (1.7), conduisant dans les deux cas à la génération d'une onde P, d'une onde S ou de la combinaison d'une onde P et d'une onde S pour des configurations isotrope et anisotrope.

1.2 Anisotropie de type TTI

1.2.1 Définition et propriétés physiques

Le tenseur d'élasticité intervenant dans le système de l'élastodynamique (1.5) est de rang quatre, c'est-à-dire qu'il est décrit par les termes $\underline{\underline{C}} = (C_{ijkl})_{1 \leq i,j,k,l \leq d}$, où d est la dimension du domaine d'étude. Il appartient à la classe des tenseurs symétriques, ce qui permet de le réécrire sous forme matricielle, suivant la notation de Voigt selon laquelle les indices sont regroupés deux à deux et remplacés par la table de correspondance Tab. 1.1.

indice tensoriel $\underline{\underline{C}}$	11	22	33	23/32	13/31	12/21
indice matriciel $\underline{\underline{C}}$	1	2	3	4	5	6

TABLE 1.1 – Correspondance des indices entre tenseur et matrice d'élasticité

Ainsi, chaque coefficient tensoriel $(C_{ijkl})_{1 \leq i,j,k,l \leq d}$ peut se réécrire comme un coefficient matriciel $(C_{mn})_{1 \leq m,n \leq 2d}$. La matrice C obtenue est, elle aussi, symétrique, composée d'au plus 21 coefficients indépendants. De plus, les paramètres physiques assurent que le tenseur est positif, i.e. $\forall \underline{\underline{\xi}}$ tenseur symétrique, $(\underline{\underline{C}} \underline{\underline{\xi}}) : \underline{\underline{\xi}} > 0$, où le symbole “:” désigne le produit scalaire tensoriel.

Matrices d'élasticité isotropes et anisotropes

Lorsqu'un milieu est isotrope, la matrice d'élasticité se simplifie considérablement avec seulement 2 constantes indépendantes. Les paramètres physiques sont généralement donnés sous la forme des constantes de Lamé (λ, μ) , correspondant respectivement au coefficient de Poisson et au module de Young. La matrice s'écrit alors :

$$C_{iso(\lambda,\mu)} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ & & \lambda + 2\mu & 0 & 0 & 0 \\ & & & \mu & 0 & 0 \\ & & & & \mu & 0 \\ & & & & & \mu \end{bmatrix} \quad (1.17)$$

Un milieu isotrope peut également être décrit par sa masse volumique ρ et les vitesses de propagation V_p et V_s relatives aux ondes P et aux ondes S qui ont été définies en introduction. Dans ce cas, la matrice s'écrit :

$$C_{iso} = \begin{bmatrix} C_{11} & C_{11} - 2C_{66} & C_{11} - 2C_{66} & 0 & 0 & 0 \\ & C_{11} & C_{11} - 2C_{66} & 0 & 0 & 0 \\ & & C_{11} & 0 & 0 & 0 \\ & & & C_{66} & 0 & 0 \\ & & & & C_{66} & 0 \\ & & & & & C_{66} \end{bmatrix} \quad (1.18)$$

où les coefficients $C_{11} = \rho V_p^2$ et $C_{66} = \rho V_s^2$ sont directement reliés à la vitesse des ondes P et S.

Isotropie transverse verticale

Dans une couche géologique, les vitesses de propagation des ondes sont en général non-uniformes dans toutes les directions mais leurs répartitions, c'est-à-dire leurs fronts d'onde, admettent deux plans de symétrie orthogonaux. C'est la définition de l'orthotropie. Si les axes du repère d'étude sont colinéaires à ces plans, la matrice d'élasticité s'écrit alors grâce à 9 constantes indépendantes :

$$C_{ortho} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ & C_{22} & C_{23} & 0 & 0 & 0 \\ & & C_{33} & 0 & 0 & 0 \\ & & & C_{44} & 0 & 0 \\ & & & & C_{55} & 0 \\ & & & & & C_{66} \end{bmatrix} \quad (1.19)$$

Si ce repère d'orthotropie est différent du repère d'étude, il est toujours possible de former la matrice d'élasticité après rotation sur le tenseur, comme cela est expliqué plus loin.

L'anisotropie du sous-sol est considérée orthotrope dans une seule direction, cela signifie que les fronts d'onde admettent une symétrie par rapport à un axe (souvent proche de la verticale). C'est la définition de l'anisotropie polaire ou Isotropie Transverse (TI), largement utilisée en imagerie sismique. Dans ce cas, le milieu est isotrope dans le plan orthogonal à cet axe, c'est-à-dire que les vitesses de propagation sont invariantes dans ce plan. Sur ce point, nous renvoyons au livre de Tsvankin [Tsv01], ou pour une version plus condensée à l'article [Tsv96]. La Fig. 1.2 représente schématiquement les différences entre des matériaux isotrope, TI et orthotrope. La Fig. 1.3 est un exemple de couches géologiques TI.

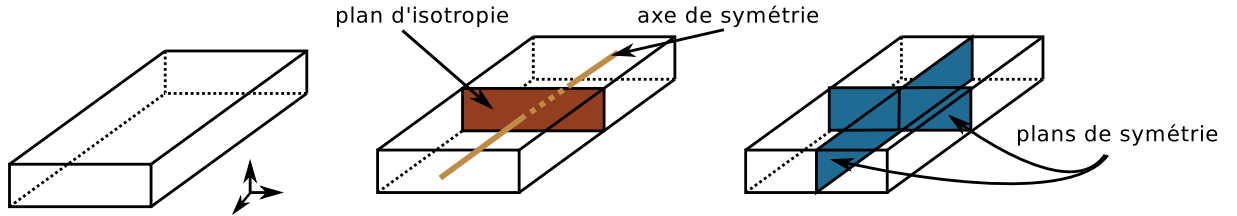


FIGURE 1.2 – Schémas de matériaux isotrope (gauche), TI (milieu) et orthotrope (droite)

Dans un repère orthogonal où l'axe \mathbf{e}_z correspond à l'axe de symétrie $\tilde{\mathbf{e}}_z$, l'anisotropie polaire est dite VTI (*Vertical Transverse Isotropy*) et la matrice d'élasticité s'écrit avec 5 constantes indépendantes :

$$C_{VTI} = \begin{bmatrix} C_{11} & C_{11} - 2C_{66} & C_{13} & 0 & 0 & 0 \\ & C_{11} & C_{13} & 0 & 0 & 0 \\ & & C_{33} & 0 & 0 & 0 \\ & & & C_{44} & 0 & 0 \\ & & & & C_{44} & 0 \\ & & & & & C_{66} \end{bmatrix} \quad (1.20)$$

Les coefficients de la matrice d'élasticité VTI (1.20) sont construits à partir de la masse volumique, des vitesses de propagation des ondes données dans la direction d'orthotropie et



FIGURE 1.3 – Photographie de couches géologiques TI, Grand Canyon USA (crédits Luca Galuzzi - www.galuzzi.it)

de trois constantes ε , δ et γ :

$$\begin{cases} C_{11} &= \rho V_p^2 (1 + 2\varepsilon) \\ C_{33} &= \rho V_p^2 \\ C_{13} &= \rho \sqrt{(V_p^2 - V_s^2)^2 + 2V_p^2 \delta (V_p^2 - V_s^2)} - \rho V_s^2 \\ C_{44} &= \rho V_s^2 \\ C_{66} &= \rho V_s^2 (1 + 2\gamma) \end{cases} \quad (1.21)$$

Ces constantes, introduites par Thomsen dans [Tho86], définissent le caractère isotrope transverse d'un matériau. Elles sont définies par les formules :

$$\begin{cases} \varepsilon &= \frac{C_{11} - C_{33}}{2C_{33}} \\ \delta &= \frac{(C_{13} + C_{44})^2 - (C_{33} - C_{44})^2}{2C_{33}(C_{33} + C_{44})} \\ \gamma &= \frac{C_{66} - C_{44}}{2C_{44}} \end{cases} \quad (1.22)$$

Nous utiliserons par la suite la notation $\kappa = \sqrt{1 + 2\varepsilon}$ pour faciliter la lisibilité des calculs.

Remarque 1.1 *L'anisotropie TI est caractérisée elliptique lorsque $\delta = \varepsilon$.*

Remarque 1.2 *Lorsque les constantes de Thomsen sont nulles (ie $\varepsilon = \delta = \gamma = 0$), la matrice d'élasticité VTI (1.20) redevient isotrope, c'est-à-dire qu'elle s'écrit comme en (1.18).*

Isotropie transverse inclinée

En général, le repère d'étude cartésien $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ ne coïncide pas avec le repère d'orthotropie $(\widetilde{\mathbf{e}}_x, \widetilde{\mathbf{e}}_y, \widetilde{\mathbf{e}}_z)$. En d'autres termes, l'axe de symétrie de l'anisotropie polaire n'est pas l'axe vertical \mathbf{e}_z . Dans ce cas, l'isotropie transverse est dite inclinée et se note TTI selon son appellation anglaise *Tilted Transverse Isotropy*. L'inclinaison est caractérisée par deux angles, représentés sur la Fig. 1.4. Plus précisément, en notant P_{xy} la projection sur le plan d'axes $(\mathbf{e}_x, \mathbf{e}_y)$, les angles sont définis par :

$$\begin{cases} \theta &= \widehat{(\mathbf{e}_z, \widetilde{\mathbf{e}}_z)} \\ \phi &= \widehat{(\mathbf{e}_x, P_{xy}\widetilde{\mathbf{e}}_z)} \end{cases} \quad (1.23)$$

où θ correspond à l'angle d'inclinaison par rapport à la vertical, appelé *dip* ou *tilt* par les géophysiciens, et ϕ est l'azimut (c'est-à-dire l'angle de la projection dans le plan horizontal).

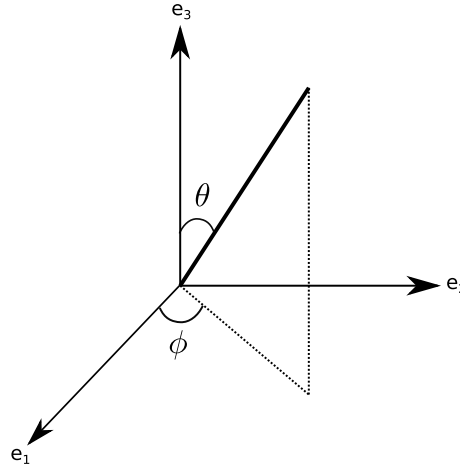


FIGURE 1.4 – Angles d'inclinaison TTI

Remarque 1.3 En géophysique, l'axe vertical \mathbf{e}_z est usuellement orienté positivement vers le bas pour suivre les valeurs croissantes de la profondeur du sous-sol.

Une propriété intéressante de l'anisotropie TTI est qu'elle peut être caractérisée algébriquement comme une rotation de l'anisotropie VTI. Cependant, comme les matrices d'élasticité sont des réécritures de tenseurs, la rotation se fait dans le cadre de l'algèbre tensorielle. Dans ce cas, deux représentations du tenseur TTI sont possibles. Soit l'inclinaison est exprimée *via* les angles (1.23) comme une matrice de passage $M_{(\theta, \phi)}$ liant le tenseur VTI et le tenseur TTI. Soit cette inclinaison est représentée grâce à une matrice de rotation $R_{(\theta, \phi)}$. Dans le premier cas, la matrice d'élasticité TTI s'obtient comme le produit $M_{(\theta, \phi)} C_{VTI} M_{(\theta, \phi)}^T$ et nous renvoyons à l'annexe A.1 pour l'expression de $M_{(\theta, \phi)}$. Dans le second cas, la matrice du changement de base entre le repère cartésien et le repère TTI $R_{(\theta, \phi)}$ est donnée par :

$$R_{(\theta, \phi)} = \begin{bmatrix} \cos \theta \cos \phi & \cos \theta \sin \phi & -\sin \theta \\ -\sin \phi & \cos \phi & 0 \\ \sin \theta \cos \phi & \sin \theta \sin \phi & \cos \theta \end{bmatrix} \quad (1.24)$$

et les coefficients $\underline{\underline{\tilde{C}}}$ du tenseur d'élasticité TTI se calculent à l'aide des coefficients $\underline{\underline{C}}$ du tenseur d'élasticité VTI selon la formule :

$$\underline{\underline{\tilde{C}}}_{ijkl} = \sum_{p=1}^d \sum_{q=1}^d \sum_{r=1}^d \sum_{s=1}^d R_{pi} R_{qj} R_{rk} R_{sl} \underline{\underline{C}}_{pqrs}, \quad \forall 1 \leq i, j, k, l \leq d \quad (1.25)$$

La matrice d'élasticité TTI ainsi obtenue est dense, composée de 21 coefficients différents. Bien que ne dépendant que de 7 constantes : $V_p, V_s, \varepsilon, \delta, \gamma, \theta, \phi$ et de la masse volumique ρ fournie par le problème physique, ces coefficients n'ont pas de forme analytique simple. Ainsi, la matrice d'élasticité TTI s'écrit par commodité :

$$C_{TTI} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ & & C_{33} & C_{34} & C_{35} & C_{36} \\ & & & C_{44} & C_{45} & C_{46} \\ & & & & C_{55} & C_{56} \\ & & & & & C_{66} \end{bmatrix} \quad (1.26)$$

La structure dense de la matrice d'élasticité TTI (1.26) est comparable à celle de l'anisotropie totale, contrairement à l'orthotropie non-inclinée (et donc à l'anisotropie VTI (1.20)) qui ne se traduit que par une modification des coefficients de la matrice d'élasticité isotrope (1.18). Même si les caractérisations des anisotropies VTI et TTI sont assez proches, l'anisotropie TTI est un problème bien plus difficile à traiter.

Remarque 1.4 Lorsque $\theta = \phi = 0$ et donc qu'il n'y a pas d'inclinaison, la matrice d'élasticité TTI (1.26) redevient VTI, c'est-à-dire comme illustrée en (1.20).

Remarque 1.5 Lorsque les constantes de Thomsen sont nulles i.e. $\varepsilon = \delta = \gamma = 0$, la matrice d'élasticité TTI (1.26) redevient isotrope, c'est-à-dire comme (1.18), quels que soient les angles d'inclinaison. Ceci est dû au fait que dans un matériau isotrope, les vitesses de propagation sont les mêmes dans toutes les directions.

Fronts d'ondes

Pour illustrer les différences physiques entre l'isotropie et l'anisotropie de type TI, nous avons représenté à la Fig. 1.5 les fronts d'ondes en deux dimensions pour un exemple de matériau élastique.

Les couches du sous-sol sont décrites par des constantes TI (1.22) très petites, toujours inférieures à un. Pour plus de détails, une liste non-exhaustive de matériaux TI est disponible dans l'article [Tho86]. Dans ce type de configurations, le front des ondes P est quasi-ellipsoïdal, ce qui est caractéristique de l'anisotropie polaire. À l'opposé, le front des ondes S peut s'avérer beaucoup plus singulier. Selon [TT94], sa forme est fortement déterminée par le paramètre :

$$\varsigma = \frac{V_p^2}{V_s^2}(\varepsilon - \delta) \quad (1.27)$$

La Fig. 1.6 représente différents fronts d'ondes S dans un matériau élastique, en faisant varier le paramètre ς .

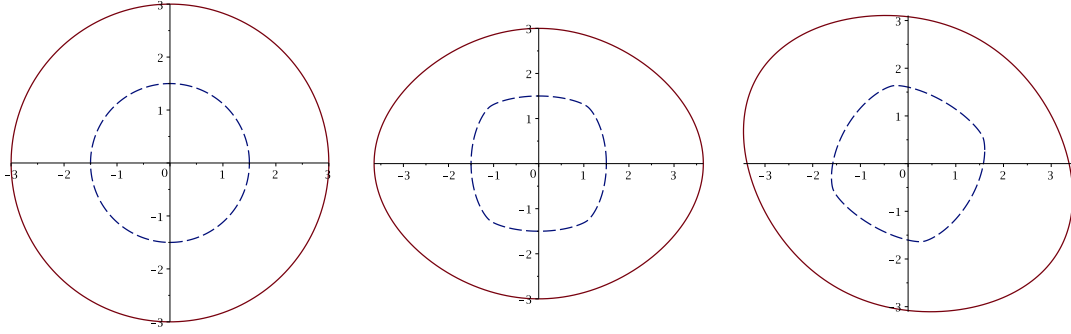


FIGURE 1.5 – Fronts d'ondes élastiques en dimension 2 : isotropes (à gauche), VTI (au milieu) et TTI (à droite), avec $\rho = 1 \text{ kg.m}^{-2}$, $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1500 \text{ m.s}^{-1}$, $\varepsilon = 0.24$, $\delta = 0.10$ et $\theta = 30^\circ$. Fronts des ondes P en rouge trait plein et des ondes S en bleu trait pointillé

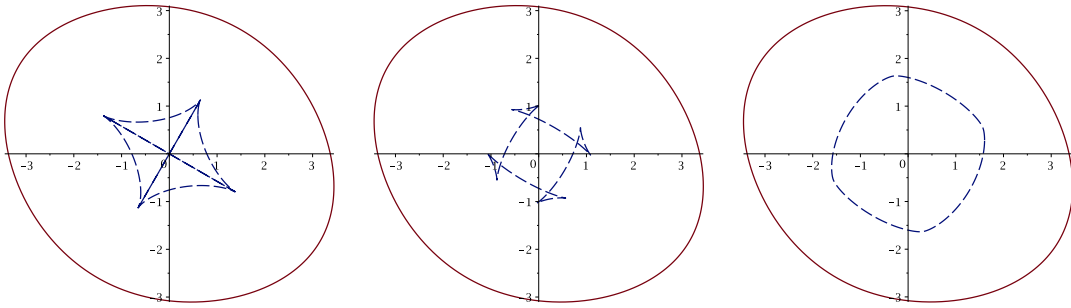


FIGURE 1.6 – Fronts d'ondes élastiques en dimension 2 : $\varsigma = \infty$ (à gauche), $\varsigma = 1.75$ (au milieu) et $\varsigma = 0.75$ (à droite), avec $\rho = 1 \text{ kg.m}^{-2}$, $V_p = 3000 \text{ m.s}^{-1}$, $\varepsilon = 0.24$, $\delta = 0.10$ et $\theta = 30^\circ$. Fronts des ondes P en rouge trait plein et des ondes S en bleu trait pointillé

1.2.2 Equations de l'élastodynamique dans un milieu anisotrope

Pour chacune des configurations, isotrope, VTI ou TTI, les équations de l'élastodynamique diffèrent. Pour faciliter le traitement des équations, il convient de développer le système général (1.5) selon chaque composante des champs d'ondes. Pour les matériaux isotropes, le système s'écrit le plus souvent avec la matrice d'élasticité (1.17) :

$$\left\{ \begin{array}{l} \rho \partial_t v_x = \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} \\ \rho \partial_t v_y = \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + \partial_z \sigma_{yz} \\ \rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_y \sigma_{yz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} = (\lambda + 2\mu) \partial_x v_x + \lambda \partial_y v_y + \lambda \partial_z v_z \\ \partial_t \sigma_{yy} = \lambda \partial_x v_x + (\lambda + 2\mu) \partial_y v_y + \lambda \partial_z v_z \\ \partial_t \sigma_{zz} = \lambda \partial_x v_x + \lambda \partial_y v_y + (\lambda + 2\mu) \partial_z v_z \\ \partial_t \sigma_{yz} = \mu (\partial_y v_z + \partial_z v_y) \\ \partial_t \sigma_{xz} = \mu (\partial_x v_z + \partial_z v_x) \\ \partial_t \sigma_{xy} = \mu (\partial_x v_y + \partial_y v_x) \end{array} \right. \quad (1.28)$$

Dans le cas des matériaux VTI, quelques coefficients sont modifiés pour former la matrice

d'élasticité (1.20) et le système général (1.5) s'écrit :

$$\left\{ \begin{array}{l} \rho \partial_t v_x = \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} \\ \rho \partial_t v_y = \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + \partial_z \sigma_{yz} \\ \rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_y \sigma_{yz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} = C_{11} \partial_x v_x + (C_{11} - 2C_{66}) \partial_y v_y + C_{13} \partial_z v_z \\ \partial_t \sigma_{yy} = (C_{11} - 2C_{66}) \partial_x v_x + C_{11} \partial_y v_y + C_{13} \partial_z v_z \\ \partial_t \sigma_{zz} = C_{13} \partial_x v_x + C_{13} \partial_y v_y + C_{33} \partial_z v_z \\ \partial_t \sigma_{yz} = C_{44} (\partial_y v_z + \partial_z v_y) \\ \partial_t \sigma_{xz} = C_{44} (\partial_x v_z + \partial_z v_x) \\ \partial_t \sigma_{xy} = C_{66} (\partial_x v_y + \partial_y v_x) \end{array} \right. \quad (1.29)$$

Pour les matériaux TTI, la matrice d'élasticité (1.26) est dense. Les équations portant sur le tenseur des contraintes contiennent donc des termes supplémentaires, qui ne sont pas présents en isotrope ou en VTI :

$$\left\{ \begin{array}{l} \rho \partial_t v_x = \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} \\ \rho \partial_t v_y = \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + \partial_z \sigma_{yz} \\ \rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_y \sigma_{yz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} = C_{11} \partial_x v_x + C_{12} \partial_y v_y + C_{13} \partial_z v_z + C_{14} (\partial_y v_z + \partial_z v_y) + C_{15} (\partial_x v_z + \partial_z v_x) + C_{16} (\partial_x v_y + \partial_y v_x) \\ \partial_t \sigma_{yy} = C_{12} \partial_x v_x + C_{22} \partial_y v_y + C_{23} \partial_z v_z + C_{24} (\partial_y v_z + \partial_z v_y) + C_{25} (\partial_x v_z + \partial_z v_x) + C_{26} (\partial_x v_y + \partial_y v_x) \\ \partial_t \sigma_{zz} = C_{13} \partial_x v_x + C_{23} \partial_y v_y + C_{33} \partial_z v_z + C_{34} (\partial_y v_z + \partial_z v_y) + C_{35} (\partial_x v_z + \partial_z v_x) + C_{36} (\partial_x v_y + \partial_y v_x) \\ \partial_t \sigma_{yz} = C_{14} \partial_x v_x + C_{24} \partial_y v_y + C_{34} \partial_z v_z + C_{44} (\partial_y v_z + \partial_z v_y) + C_{45} (\partial_x v_z + \partial_z v_x) + C_{46} (\partial_x v_y + \partial_y v_x) \\ \partial_t \sigma_{xz} = C_{15} \partial_x v_x + C_{25} \partial_y v_y + C_{35} \partial_z v_z + C_{45} (\partial_y v_z + \partial_z v_y) + C_{55} (\partial_x v_z + \partial_z v_x) + C_{56} (\partial_x v_y + \partial_y v_x) \\ \partial_t \sigma_{xy} = C_{16} \partial_x v_x + C_{26} \partial_y v_y + C_{36} \partial_z v_z + C_{46} (\partial_y v_z + \partial_z v_y) + C_{56} (\partial_x v_z + \partial_z v_x) + C_{66} (\partial_x v_y + \partial_y v_x) \end{array} \right. \quad (1.30)$$

Cette différence structurelle dans le cas d'une anisotropie de type TTI entraîne des complications algébriques évidentes, du fait de l'apparition de termes supplémentaires dans les équations. Cependant, cette complexité est essentiellement algébrique et en observant avec soin comment est défini le système dans un matériau TTI, il apparaît que les coefficients de la matrice d'élasticité (1.26) ne sont pas indépendants, contrairement au cas d'anisotropie totale. Ainsi, l'anisotropie de type TTI peut être décrite comme un cas d'anisotropie VTI ayant subi une inclinaison. Ce point va être déterminant pour nous, et nous utiliserons cette propriété plus loin quand il s'agira de développer des conditions de bord absorbant.

1.3 Discrétisation des équations

1.3.1 Définition et état de l'art

La méthode de Galerkin, apparue dans les années 1910, sert à formuler le problème discret associé à la résolution des systèmes d'équations aux dérivées partielles. Le principe repose sur un choix de sous-espace de fonctions de dimension finie permettant de décrire dans le domaine discret les fonctions du problème continu. Elle est le fondement des méthodes d'éléments finis (FEM pour *Finite Element Method*).

La méthode de Galerkin discontinue, très connue sous son acronyme DGM pour sa terminologie anglaise (*Discontinuous Galerkin Method*) est plus récente, datant des années 1970. Quelques fois décrite comme une FEM avancée, son principe ajoute au choix du sous-espace fini de fonctions discrètes la possibilité de travailler à l'échelle locale de l'élément. Les fonctions de base sont alors propres à chaque cellule du maillage ce qui signifie qu'elles sont discontinues à l'interfaces entre deux éléments (voir une illustration à la Fig. 1.7).

La DGM, tout comme la FEM, permet de considérer des maillages complexes et de procéder à des approximations d'ordre élevé. La distribution des degrés de libertés est rappelée sur les Fig. 1.8 et tableau 1.2. La DGM a pour avantage de conduire à une formulation temporelle des champs quasi-explicite car la matrice de masse est diagonale par blocs grâce à la discontinuité des fonctions de base. De plus, la largeur de bande de la matrice de rigidité est la même pour tous les éléments, car un élément ne communique avec ses voisins que par les faces alors qu'avec la FEM il communique *via* les faces, les arêtes et les sommets. La matrice de rigidité possède donc une structure par blocs. Si le maillage contient N mailles, elle sera composée de N blocs diagonaux et de $(d + 1)N$ blocs extradiagonaux traduisant les communications inter-éléments. La taille des blocs dépend uniquement de l'ordre de la discrétisation. Cette structure se prête donc très facilement à une implémentation dans un cadre HPC.

Une autre caractéristique de la DGM est l'adaptabilité *hp*, c'est-à-dire qu'en autorisant la discontinuité des fonctions de base, il est possible d'utiliser des maillages non-conformes (adaptabilité *h*) et mixer des ordres d'approximation différents (adaptabilité *p*), ce qui assure une plus grande flexibilité de la méthode, voir Fig. 1.9. En pratique, c'est l'adaptabilité *p* qui est utilisée dans le code DIVA car l'adaptabilité *h* présente plusieurs inconvénients de mise en œuvre. Tout d'abord elle nécessite de changer considérablement la structure du maillage pour prendre en compte les *hanging nodes* (i.e. les nœuds d'un élément situés sur une arête ou une face d'un autre élément), alors que l'adaptabilité *p* n'a besoin que d'un vecteur supplémentaire indiquant l'ordre de chaque élément. Le stockage de vecteur peut même être évité en réordonnant judicieusement les éléments en fonction de leur ordre de discrétisation. Ensuite, l'adaptabilité *h* complexifie considérablement les calculs d'intégrales de surface, notamment en 3D. En effet, il n'est plus possible de se ramener à un calcul sur un élément de référence car les calculs doivent être faits sur une partie de la face correspondant à l'intersection de deux éléments et non sur une face entière. En 3D, l'intersection de deux tétraèdres d'un maillage non-conforme n'est pas forcément un triangle mais peut être un quadrilatère, un pentagone ou un hexagone. L'adaptabilité *p* ne nécessite que le calcul des différentes combinaisons d'ordre possibles sur l'élément de référence. En pratique, le nombre de ces combinaisons est limité, de l'ordre d'une dizaine. Finalement, l'adaptabilité *h* casse la structure de la matrice de raideur (qui sera présentée plus loin) car le nombre de voisins d'un élément devient variable. L'adaptabilité *p* conserve cette structure et modifie uniquement la taille des blocs.

La DGM a été introduite par Reed et Hill dans [RH73] tout d'abord pour les équations elliptiques puis paraboliques. Citons [ABCM02] notamment pour sa bibliographie détaillée. Pour les équations hyperboliques, une première analyse mathématique est donnée par Lesaint et Raviart au milieu des années 1970 [LR74, Les75] et l'ordre de convergence optimal a été établi par Nävert et Pitkäranta dix ans plus tard [Näv82, JNP84, JP86]. La DGM est désormais populaire pour les équations hyperboliques (linéaires et non-linéaires)

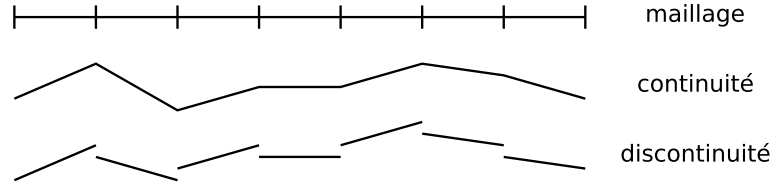


FIGURE 1.7 – Fonctions discrètes sur un maillage à une dimension, continues et discontinues

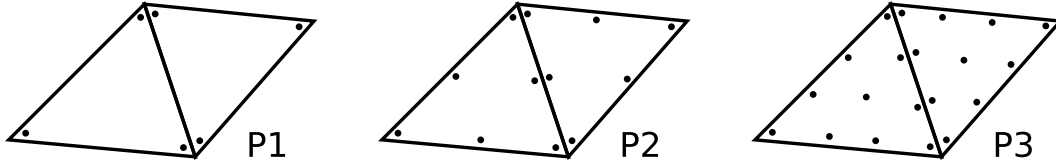
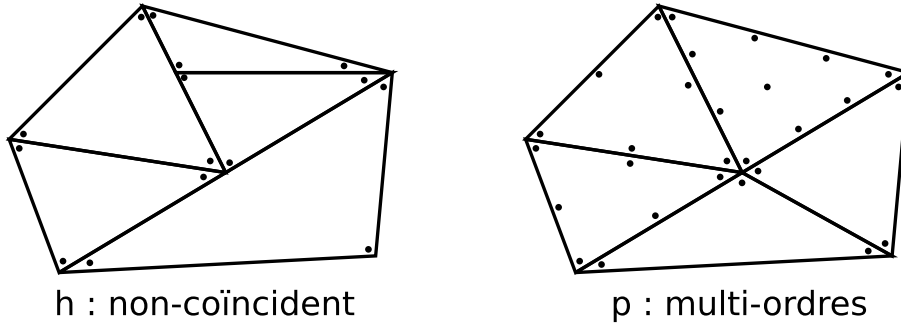


FIGURE 1.8 – Localisation des degrés de liberté en deux dimensions, pour trois discrétisations

	2D			3D		
	P1	P2	P3	P1	P2	P3
volume : K	3	6	10	4	10	20
surface : F	2	3	4	3	6	10

TABLE 1.2 – Nombre de degrés de liberté en deux et trois dimensions


FIGURE 1.9 – Illustration de l'adaptabilité hp sur un maillage en deux dimension

et les travaux de Cockburn, Shu et leurs co-auteurs ont été particulièrement convaincants. Dans [CKS00a, Coc01], l'intégration en temps est menée en appliquant une méthode de Runge-Kutta et ces travaux sont bien connus dans la communauté sous l'acronyme RKDG. Un état de l'art des approches DG jusqu'aux années 2000 est disponible dans [CKS00b]. En ce qui concerne l'adaptabilité hp , les travaux de Süli, Schwab et Houston dans [SHS00, SSH00] sont particulièrement déterminants.

De nos jours, et particulièrement pour la propagation des ondes, plusieurs équipes utilisent la DGM et les approches diffèrent essentiellement au niveau des flux et des schémas en temps. Dumbser, Käser et leurs co-auteurs privilégient le schéma ADER (*Arbitrary high-order DERivation*) basé sur des flux décentrés avant et des intégrations en temps de type Runge-Kutta (voir par exemple [DK06, KD06, KHDIP08]). Leur méthode, appelée DG-ADER, a l'inconvénient d'être dissipative. D'autres équipes, comme [BLP06, DFG009], préfèrent ap-

plier le schéma Leap-Frog avec des flux centrés, offrant ainsi une alternative non-dissipative. Indépendamment, des techniques de pénalités intérieures ont été développées pour les équations en formulation du second ordre, comme la IPDGM (*Interior Penalty DGM*). Pour les équations hyperboliques, citons par exemple [GSS06, AMM06, BSW08, Bal09].

Nous utilisons la formulation du premier ordre de l'équation des ondes et nous avons choisi le schéma de discrétisation DG avec flux centrés en espace et Leap-Frog en temps. Pour le schéma en temps une itération de Leap-Frog est moins coûteuse qu'une itération de Runge-Kutta. Pour le schéma en espace, l'utilisation de flux centrés peut entraîner l'apparition d'ondes parasites ainsi qu'une détérioration de l'ordre de convergence du schéma. L'idéal serait donc de coupler le schéma Leap-Frog et DG avec flux décentrés avant. Malheureusement, un tel couplage conduit à un schéma implicite en temps, plus complexe à mettre en œuvre dans un cadre de résolution parallèle et trop coûteux pour des applications à l'échelle industrielle. Il est cependant possible de supprimer les ondes parasites et de conserver l'ordre de convergence optimale du schéma Leap-Frog et DG avec flux centrés en lui ajoutant un terme de pénalisation non-dissipatif [Ven14], au prix d'une diminution de la condition CFL.

1.3.2 Discrétisation en espace

Soit Ω_h une discrétisation du domaine Ω en cellules polygonales K , aussi proche que possible d'une partition de Ω , avec $\Omega_h \subset \Omega$. On notera \mathcal{T}_h l'ensemble des cellules K . La frontière Γ_h est décomposée en partie extérieure $\Gamma_{out} = \partial\Omega_h$ et intérieure Γ_{in} , correspondant à l'ensemble des faces entre les cellules. Sur un élément K , nous noterons \mathbf{n}_K la normale à ∂K sortante.

Toute face interne est partagée par un élément K_1 et un élément K_2 . Nous noterons arbitrairement $\mathbf{n} = \mathbf{n}_{K_1}$ et nous définissons le symbole $[[\mathbf{u}]] \cdot \mathbf{n} = (\mathbf{u}_{K_2} - \mathbf{u}_{K_1}) \cdot \mathbf{n}$ comme le saut et le symbole $\{\{\mathbf{u}\}\} = \frac{1}{2}(\mathbf{u}_{K_1} + \mathbf{u}_{K_2})$ comme la moyenne. Pour une face externe, $\mathbf{n} = \mathbf{n}_K$ et par convention le saut et la moyenne sur cette face sont égaux à l'identité.

Formulation variationnelle

Soient les espaces :

$$V = \{\mathbf{v} \in L^2(\Omega_h)^d \text{ tel que } \mathbf{v}|_K \in H^1(\Omega_h)^d, \forall K \in \mathcal{T}_h\}$$

$$\Sigma = \{\underline{\underline{\sigma}} \in L^2(\Omega_h)^{d^2} \text{ tel que } \underline{\underline{\sigma}}|_K \in H_{\text{div}}(\Omega_h)^{d^2}, \forall K \in \mathcal{T}_h \text{ et } \sigma_{ij} = \sigma_{ji}, \forall i, j = x, y, z\}$$

La première étape consiste à écrire la formulation variationnelle associée au système de l'élastodynamique (1.5). Pour cela, nous définissons $\mathbf{v} \in V$ et $\underline{\underline{\sigma}} \in \Sigma$. Soient une fonction-test $\mathbf{w} \in V$ et un tenseur-test $\underline{\underline{\xi}} \in \Sigma$. L'intégration des équations (1.5) contre ce couple-test, sans détailler les dépendances sur les variables de temps et d'espace, donne sur une cellule K du maillage :

$$\begin{cases} \int_K \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} &= \int_K (\nabla \cdot \underline{\underline{\sigma}}) \cdot \mathbf{w} d\mathbf{x} \\ \int_K \partial_t \underline{\underline{\sigma}} : \underline{\underline{\xi}} d\mathbf{x} &= \int_K (\underline{\underline{C}} \cdot \underline{\underline{\epsilon}}(\mathbf{v})) : \underline{\underline{\xi}} d\mathbf{x} \end{cases} \quad (1.31)$$

Puis, en observant que $(\underline{\underline{C}} \underline{\underline{\epsilon}}(\mathbf{v})) : \underline{\underline{\xi}} = (\underline{\underline{C}} \underline{\underline{\xi}}) : \nabla \mathbf{v}$, l'intégration par partie de chaque équation de (1.31) donne :

$$\begin{cases} \int_K \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} &= \int_{\partial K} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{w} d\mathbf{x} - \int_K \underline{\underline{\sigma}} : \nabla \mathbf{w} d\mathbf{x} \\ \int_K \partial_t \underline{\underline{\sigma}} : \underline{\underline{\xi}} d\mathbf{x} &= \int_{\partial K} ((\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{n}) \cdot \mathbf{v} d\mathbf{x} - \int_K (\nabla \cdot (\underline{\underline{C}} \underline{\underline{\xi}})) \cdot \mathbf{v} d\mathbf{x} \end{cases} \quad (1.32)$$

Pour obtenir la formulation variationnelle, il reste à sommer pour toutes les cellules du maillage le système (1.32). Ce faisant, les termes des frontières internes Γ_{in} font apparaître les flux. Les opérateurs gradient et divergence n'étant définis que sur chaque cellule, introduisons les opérateurs brisés : ∇_h tel que $\nabla_h \mathbf{u}|_K = \nabla \mathbf{u} \forall K \in \mathcal{T}_h$ et $\nabla_h \cdot$ tel que $\nabla_h \cdot \mathbf{u}|_K = \nabla \cdot \mathbf{u} \forall K \in \mathcal{T}_h$. La somme s'écrit alors :

$$\begin{cases} \int_{\Omega_h} \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} &= \int_{\Gamma_{in}} [\underline{\underline{\sigma}} \mathbf{w}] \cdot \mathbf{n} d\mathbf{x} + \int_{\Gamma_{out}} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{w} d\mathbf{x} - \int_{\Omega_h} \underline{\underline{\sigma}} : \nabla_h \mathbf{w} d\mathbf{x} \\ \int_{\Omega_h} \partial_t \underline{\underline{\sigma}} : \underline{\underline{\xi}} d\mathbf{x} &= \int_{\Gamma_{in}} [(\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{v}] \cdot \mathbf{n} d\mathbf{x} + \int_{\Gamma_{out}} ((\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{n}) \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega_h} (\nabla_h \cdot (\underline{\underline{C}} \underline{\underline{\xi}})) \cdot \mathbf{v} d\mathbf{x} \end{cases} \quad (1.33)$$

Les termes de flux peuvent se réécrire :

$$\begin{cases} [\underline{\underline{\sigma}} \mathbf{w}] \cdot \mathbf{n} &= [\underline{\underline{\sigma}}] \{\{\mathbf{w}\}\} \cdot \mathbf{n} + \{\{\underline{\underline{\sigma}}\}\} [\mathbf{w}] \cdot \mathbf{n} \\ [(\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{v}] \cdot \mathbf{n} &= [(\underline{\underline{C}} \underline{\underline{\xi}})] \{\{\mathbf{v}\}\} \cdot \mathbf{n} + \{\{(\underline{\underline{C}} \underline{\underline{\xi}})\}\} [\mathbf{v}] \cdot \mathbf{n} \end{cases} \quad (1.34)$$

Ce qui par la continuité de \mathbf{v} et $\underline{\underline{\sigma}}$ se simplifie :

$$\begin{cases} [\underline{\underline{\sigma}} \mathbf{w}] \cdot \mathbf{n} &= \{\{\underline{\underline{\sigma}}\}\} [\mathbf{w}] \cdot \mathbf{n} \\ [(\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{v}] \cdot \mathbf{n} &= [(\underline{\underline{C}} \underline{\underline{\xi}})] \{\{\mathbf{v}\}\} \cdot \mathbf{n} \end{cases} \quad (1.35)$$

La formulation variationnelle s'écrit alors : *Trouver $(\mathbf{v}, \underline{\underline{\sigma}})$ dans $V \times \Sigma$ tel que pour tout couple-test $(\mathbf{w}, \underline{\underline{\xi}})$ dans le même espace, on ait :*

$$\begin{cases} \int_{\Omega_h} \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} &= \int_{\Gamma_{in}} \{\{\underline{\underline{\sigma}}\}\} [\mathbf{w}] \cdot \mathbf{n} d\mathbf{x} + \int_{\Gamma_{out}} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{w} d\mathbf{x} - \int_{\Omega_h} \underline{\underline{\sigma}} : \nabla_h \mathbf{w} d\mathbf{x} \\ \int_{\Omega_h} \partial_t \underline{\underline{\sigma}} : \underline{\underline{\xi}} d\mathbf{x} &= \int_{\Gamma_{in}} [(\underline{\underline{C}} \underline{\underline{\xi}})] \{\{\mathbf{v}\}\} \cdot \mathbf{n} d\mathbf{x} + \int_{\Gamma_{out}} ((\underline{\underline{C}} \underline{\underline{\xi}}) \mathbf{n}) \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega_h} (\nabla_h \cdot (\underline{\underline{C}} \underline{\underline{\xi}})) \cdot \mathbf{v} d\mathbf{x} \end{cases} \quad (1.36)$$

La formulation (1.36) en l'état conduit à un problème mal posé, il faut définir des conditions aux limites. De telles conditions modifient les intégrales sur les frontières extérieures Γ_{out} . La méthode classique consiste à remplacer dans la première intégrale de bords le terme $\underline{\underline{\sigma}} \mathbf{n}$ par une combinaison linéaire sur les composantes de \mathbf{v} , selon la condition aux limites choisie (définissant des coefficients b_{ij} , $i = 1..d$) et à laisser la seconde intégrale de bords inchangée. Par exemple, la formule la plus simple $\underline{\underline{\sigma}} \mathbf{n} = 0$ est une condition de Neumann. Nous renvoyons aux chapitre 2 et 3 pour une description détaillée des conditions aux limites absorbantes utilisées ainsi qu'une analyse de stabilité des problèmes mixtes associés.

Système matriciel semi-discret

Nous allons maintenant discrétiser l'inconnue \mathbf{v} sur le sous-espace de fonctions V_h tel que

$$V_h = \{\mathbf{v} \in L^2(\Omega_h)^d \text{ tel que } (\mathbf{v}|_K)_i \in \mathbb{P}(K)^l, \forall K \in \mathcal{T}_h, \forall i = x, y, z\}$$

où $\mathbb{P}(K)^l$ est l'ensemble des polynômes de degré inférieur ou égal à l . L'ordre de discrétisation est alors défini par le choix de ce degré l , pour l'interpolation sur les cellules. L'un des avantages majeurs des méthodes DG est la possibilité d'utiliser des degrés différents sur chaque cellule, en fonction de leur taille et de leurs caractéristiques physiques. Cependant, pour ne pas alourdir la présentation, nous supposons ici que ce degré est le même pour toutes les cellules.

L'inconnue $\underline{\underline{\sigma}}$ est discrétisée sur le sous-espace de fonctions Σ_h tel que

$$\Sigma_h = \{\underline{\underline{\sigma}} \in L^2(\Omega_h)^{d^2} \text{ tel que } (\underline{\underline{\sigma}}|_K)_{ij} \in \mathbb{P}(K)^l, \forall K \in \mathcal{T}_h \text{ et } \sigma_{ij} = \sigma_{ji}, \forall i, j = x, y, z\}$$

Grâce à la discontinuité des fonctions, nous pouvons construire une base des espaces V_h et Σ_h en procédant élément par élément. Soit $(\varphi_i^K)_{i=1..N_l}$ une base du sous-espace $\mathbb{P}(K)^l$, par exemple les fonctions de base de Lagrange. La dimension N_l de $\mathbb{P}(\mathbb{K})^l$ est $(l+1)$ en 1D, $(l+1)(l+2)/2$ en 2D et $(l+1)(l+2)(l+3)/6$ en 3D. L'ensemble des fonctions $((\varphi_i^K)_{i=1..N_l, K \in \mathcal{T}_h})$ forme une base de l'espace $V_{h|d=1}$ et une fonction c peut alors être approchée par $c_h \in V_{h|d=1}$.

$$c_h(\mathbf{x}, t) = \sum_{K \in \mathcal{T}_h} \sum_{i=1..N_l} c_i^K(t) \varphi_i^K(\mathbf{x}).$$

L'inconnue vectorielle \mathbf{v} est approchée par $\mathbf{v}_h \in V_h$,

$$\mathbf{v}_h(\mathbf{x}, t) = \sum_{K \in \mathcal{T}_h} \sum_{j=x,y,z} \sum_{i=1..N_l} v_{ji}^K(t) \varphi_i^K(\mathbf{x}) \mathbf{e}_j$$

et le tenseur $\underline{\underline{\sigma}}$ est approché par $\underline{\underline{\sigma}}_h \in \Sigma_h$,

$$\underline{\underline{\sigma}}_h(\mathbf{x}, t) = \sum_{K \in \mathcal{T}_h} \sum_{j=x,y,z} \sum_{k=x,y,z} \sum_{i=1..N_l} \sigma_{jk_i}^K(t) \varphi_i^K(\mathbf{x}) \mathbf{e}_j \mathbf{e}_k,$$

où nous imposons $\sigma_{jk_i}^K = \sigma_{kj_i}^K$ pour tous K, i, j, k afin d'assurer la symétrie du tenseur $\underline{\underline{\sigma}}_h$. Remarquons que

$$\mathbf{v}_{h|K}(\mathbf{x}, t) = \sum_{j=x,y,z} \sum_{i=1..N_l} v_{ji}^K(t) \varphi_i^K(\mathbf{x}) \mathbf{e}_j$$

et

$$\underline{\underline{\sigma}}_{h|K}(\mathbf{x}, t) = \sum_{j=x,y,z} \sum_{k=x,y,z} \sum_{i=1..N_l} \sigma_{jk_i}^K(t) \varphi_i^K(\mathbf{x}) \mathbf{e}_j \mathbf{e}_k.$$

Nous utiliserons donc par la suite une notation par blocs pour les inconnues : pour chaque élément K , pour tout $j = x, y, z$ et pour tout $k = x, y, z$, $k \geq j$, nous définissons les vecteurs $\mathbf{v}_j^K = (v_{ji}^K)_{i=1..N_l}$ et $\boldsymbol{\sigma}_{jk}^K = (\sigma_{jk_i}^K)_{i=1..N_l}$. Nous définissons ensuite les vecteurs

$$\mathbf{v}_h^K = [\mathbf{v}_x^K, \mathbf{v}_y^K, \mathbf{v}_z^K]^T \text{ et } \boldsymbol{\sigma}_h^K = [\boldsymbol{\sigma}_{xx}^K, \boldsymbol{\sigma}_{yy}^K, \boldsymbol{\sigma}_{zz}^K, \boldsymbol{\sigma}_{yz}^K, \boldsymbol{\sigma}_{xz}^K, \boldsymbol{\sigma}_{xy}^K]^T.$$

1.3. Discrétisation des équations

Finalement, \mathbf{v}_h et $\boldsymbol{\sigma}_h$ désignent respectivement les vecteurs composés de l'ensemble des vecteurs \mathbf{v}_h^K et $\boldsymbol{\sigma}_h^K$. En choisissant les fonctions test $\mathbf{w} = \varphi_i^K \mathbf{e}_j$ et $\underline{\xi} = \varphi_i^K \mathbf{e}_j \mathbf{e}_k$ pour tout $j = x, y, z$ et pour tout $k = x, y, z$, $k \geq j$, la formulation variationnelle (1.36), avec des conditions aux limites, se transforme en un système linéaire matriciel semi-discret :

$$\begin{cases} \sum_{K \in \mathcal{T}_h} \left(\rho_K M_{\mathbf{v}}^K \partial_t \mathbf{v}_h^K + R_{\boldsymbol{\sigma}}^K \boldsymbol{\sigma}_h^K + \sum_{L \in \mathcal{T}_h | \partial K \cap \partial L \neq \emptyset} R_{\boldsymbol{\sigma}}^{K,L} \boldsymbol{\sigma}_h^L + B_{\mathbf{v}}^K \mathbf{v}_h^K \right) = 0 \\ \sum_{K \in \mathcal{T}_h} \left(M_{\boldsymbol{\sigma}}^K \partial_t \boldsymbol{\sigma}_h^K + R_{\mathbf{v}}^K \mathbf{v}_h^K + \sum_{L \in \mathcal{T}_h | \partial K \cap \partial L \neq \emptyset} R_{\mathbf{v}}^{K,L} \mathbf{v}_h^L \right) = 0 \end{cases} \quad (1.37)$$

où M_{\star} désignent les matrices de masse et R_{\star} les matrices de rigidités. Comme pour les vecteurs, ces matrices peuvent être définies par blocs. Ainsi, la matrice $M_{\mathbf{v}}^K$ est une matrice de taille $dN_l * dN_l$ diagonale par d blocs $N_l * N_l$. Ces d blocs sont tous égaux à la matrice \widetilde{M}^K définie par

$$\widetilde{M}_{ij}^K = \int_K \varphi_i^K \varphi_j^K d\mathbf{x}, \quad \forall i, j = 1..N_l \quad (1.38)$$

De même, la matrice $M_{\boldsymbol{\sigma}}^K$ est une matrice de taille $d(d+1)N_l/2 * d(d+1)N_l/2$ diagonale par $d(d+1)/2$ blocs $N_l * N_l$. Ces $d(d+1)/2$ blocs sont tous égaux à la matrice \widetilde{M}^K .

La matrice $R_{\boldsymbol{\sigma}}^K$ est une matrice de taille $dN_l * d(d+1)N_l/2$ de la forme

$$R_{\boldsymbol{\sigma}}^K = \begin{bmatrix} \widetilde{R}_x^K & 0 & 0 & 0 & \widetilde{R}_z^K & \widetilde{R}_y^K \\ 0 & \widetilde{R}_y^K & 0 & \widetilde{R}_z^K & 0 & \widetilde{R}_x^K \\ 0 & 0 & \widetilde{R}_z^K & \widetilde{R}_y^K & \widetilde{R}_x^K & 0 \end{bmatrix},$$

où chaque bloc \widetilde{R}_i^K , $i = x, y, z$ est de taille $N_l * N_l$ et est défini par

$$\widetilde{R}_{ipq}^K = \int_K \varphi_p^K \frac{\partial \varphi_q^K}{\partial i} d\mathbf{x}, \quad \forall p, q = 1..N_l. \quad (1.39)$$

La matrice $R_{\mathbf{v}}^K$ est une matrice $d(d+1)N_l/2 * dN_l$ de la forme

$$R_{\mathbf{v}}^K = \begin{bmatrix} (C_{11}\widetilde{R}_x^K + C_{16}\widetilde{R}_y^K + C_{15}\widetilde{R}_z^K) & (C_{16}\widetilde{R}_x^K + C_{12}\widetilde{R}_y^K + C_{14}\widetilde{R}_z^K) & (C_{15}\widetilde{R}_x^K + C_{14}\widetilde{R}_y^K + C_{13}\widetilde{R}_z^K) \\ (C_{12}\widetilde{R}_x^K + C_{26}\widetilde{R}_y^K + C_{25}\widetilde{R}_z^K) & (C_{26}\widetilde{R}_x^K + C_{22}\widetilde{R}_y^K + C_{24}\widetilde{R}_z^K) & (C_{25}\widetilde{R}_x^K + C_{24}\widetilde{R}_y^K + C_{23}\widetilde{R}_z^K) \\ (C_{13}\widetilde{R}_x^K + C_{36}\widetilde{R}_y^K + C_{35}\widetilde{R}_z^K) & (C_{36}\widetilde{R}_x^K + C_{23}\widetilde{R}_y^K + C_{34}\widetilde{R}_z^K) & (C_{35}\widetilde{R}_x^K + C_{34}\widetilde{R}_y^K + C_{33}\widetilde{R}_z^K) \\ (C_{14}\widetilde{R}_x^K + C_{46}\widetilde{R}_y^K + C_{45}\widetilde{R}_z^K) & (C_{46}\widetilde{R}_x^K + C_{24}\widetilde{R}_y^K + C_{44}\widetilde{R}_z^K) & (C_{45}\widetilde{R}_x^K + C_{44}\widetilde{R}_y^K + C_{34}\widetilde{R}_z^K) \\ (C_{15}\widetilde{R}_x^K + C_{56}\widetilde{R}_y^K + C_{55}\widetilde{R}_z^K) & (C_{56}\widetilde{R}_x^K + C_{25}\widetilde{R}_y^K + C_{45}\widetilde{R}_z^K) & (C_{55}\widetilde{R}_x^K + C_{45}\widetilde{R}_y^K + C_{35}\widetilde{R}_z^K) \\ (C_{16}\widetilde{R}_x^K + C_{66}\widetilde{R}_y^K + C_{56}\widetilde{R}_z^K) & (C_{66}\widetilde{R}_x^K + C_{26}\widetilde{R}_y^K + C_{46}\widetilde{R}_z^K) & (C_{56}\widetilde{R}_x^K + C_{46}\widetilde{R}_y^K + C_{36}\widetilde{R}_z^K) \end{bmatrix},$$

Chapitre 1. Modélisation des équations d'ondes en milieu transverse isotrope

Les matrices $R_{\sigma}^{K,L}$ et $R_{\mathbf{v}}^{K,L}$ sont de dimensions respectives $dN_l * d(d+1)N_l/2$ et $d(d+1)N_l/2 * dN_l$ et ont les mêmes structures que les matrices R_{σ}^K et $R_{\mathbf{v}}^K$ où tous les termes $\widetilde{R}_{\star,i}^K$ sont maintenant des termes $\widetilde{R}_{\sigma,i}^{K,L}$ pour $R_{\sigma}^{K,L}$ et $\widetilde{R}_{\mathbf{v},i}^{K,L}$ pour $R_{\mathbf{v}}^{K,L}$, $i = x, y, z$. Ces blocs sont de taille $N_l * N_l$ et sont définis par

$$\begin{cases} \widetilde{R}_{\sigma,ipq}^{K,L} = -\frac{1}{2} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \varphi_q^K \varphi_p^L \mathbf{n}_K \cdot \mathbf{e}_i d\mathbf{x}, \quad \forall p, q = 1..N_l, \\ \widetilde{R}_{\mathbf{v},ipq}^{K,L} = -\frac{1}{2} \int_{\partial K \cap \partial L} \varphi_q^K \varphi_p^L \mathbf{n}_K \cdot \mathbf{e}_i d\mathbf{x}, \quad \forall p, q = 1..N_l. \end{cases} \quad (1.40)$$

Ils sont différents par rapport au domaine d'intégration qui contient ou qui exclu les faces de la frontière extérieure Γ_{out} .

Enfin, la matrice $B_{\mathbf{v}}^K$ associée à l'intégrale des conditions aux limites sur une face d'un élément K au contact de la frontière Γ_{out} est une matrice de taille $dN_l * dN_l$ où chaque bloc \widetilde{B}_{ij}^K , $i, j = x, y, z$ est de taille $N_l * N_l$ et est défini par

$$\widetilde{B}_{ijpq}^K = -b_{ij} \int_{\partial K \cap \Gamma_{out}} \varphi_p^K \varphi_q^K d\mathbf{x}, \quad \forall p, q = 1..N_l. \quad (1.41)$$

Chacune de ces matrices peut être calculée à partir de l'élément de référence \hat{K} , qui représente le segment $[0, 1]$ en 1D, le triangle $(0, 0); (1, 0); (0, 1)$ en 2D, voir Fig. 1.10 et le tétraèdre $(0, 0, 0); (1, 0, 0); (0, 1, 0); (0, 0, 1)$ en 3D. Nous notons F_K la fonction transformant l'élément de référence \hat{K} en l'élément K . Cette fonction est linéaire et bijective, car nous supposons que les sommets des éléments K sont ordonnés, ce qui revient à considérer que le triangle défini par les sommets A, B, C est différent du triangle déterminé par les sommets C, A, B .

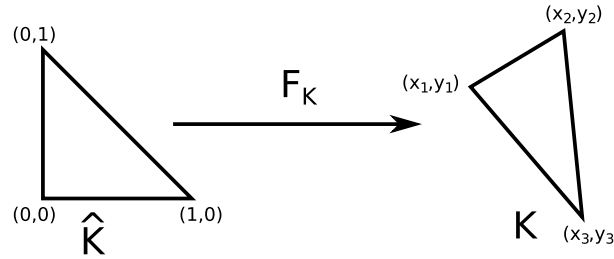


FIGURE 1.10 – Passage à l'élément de référence en 2D

Les matrices \widetilde{M}^K et \widetilde{R}_i^K s'écrivent alors :

$$\widetilde{M}_{ij}^K = \int_K \varphi_i^K \varphi_j^K d\mathbf{x} = |\det J_{F_K}| \int_{\hat{K}} \varphi_i^{\hat{K}} \varphi_j^{\hat{K}} d\mathbf{x} = |\det J_{F_K}| \widetilde{M}^{\hat{K}}.$$

où J_{F_K} est le jacobien de la transformation affine F_K , et

$$\widetilde{R}_{ipq}^K = - \int_K \varphi_q^K \frac{\partial \varphi_p^K}{\partial i} d\mathbf{x} = -|\det J_{F_K}| \int_{\hat{K}} \varphi_q^{\hat{K}} J_{F_K}^{-T} \nabla \varphi_p^{\hat{K}} \cdot \mathbf{e}_i d\mathbf{x} = -|\det J_{F_K}| \left(|J_{F_K}^{-T}| \int_{\hat{K}} \varphi_q^{\hat{K}} \nabla \varphi_p^{\hat{K}} d\mathbf{x} \right) \cdot \mathbf{e}_i.$$

Remarquons que toutes les matrices \widetilde{M}_K sont proportionnelles à la matrice $\widetilde{M}_{\hat{K}}$. Nous pouvons donc déterminer leur inverse à partir de $\widetilde{M}_{\hat{K}}^{-1}$ et de $|\det J_{F_K}|$ uniquement, ce qui permet d'accélérer considérablement les calculs.

Les matrices \widetilde{B}_{ij}^K et $\widetilde{R}_{\star,i}^{K,L}$ peuvent être calculées de la même manière sur l'élément de surface de référence.

Dans la pratique, nous résolvons le système (1.37) élément par élément, ce qui est particulièrement adapté à un cadre HPC. Cependant, pour des raisons de simplicité, nous écrirons parfois ce système sous sa forme matricielle globale

$$\begin{cases} M_{\mathbf{v}} \partial_t \mathbf{v}_h + R_{\boldsymbol{\sigma}} \boldsymbol{\sigma}_h + B_{\mathbf{v}} \mathbf{v}_h &= 0 \\ M_{\boldsymbol{\sigma}} \partial_t \boldsymbol{\sigma}_h + R_{\mathbf{v}} \mathbf{v}_h &= 0 \end{cases} \quad (1.42)$$

où les matrices $M_{\mathbf{v}}$, $R_{\boldsymbol{\sigma}}$, $B_{\mathbf{v}}$, $M_{\boldsymbol{\sigma}}$ et $R_{\mathbf{v}}$ sont définies par blocs de la façon suivante :

$$(M_{\mathbf{v}})_{K,L} = \begin{cases} \rho_K M_{\mathbf{v}}^K & \text{si } L = K, \\ 0 & \text{si } L \neq K, \end{cases} \quad (1.43)$$

$$(R_{\boldsymbol{\sigma}})_{K,L} = \begin{cases} R_{\boldsymbol{\sigma}}^K + R_{\boldsymbol{\sigma}}^{K,K} & \text{si } L = K, \\ R_{\boldsymbol{\sigma}}^{K,L} & \text{si } L \neq K \text{ et } \partial L \cup \partial K \neq \emptyset, \\ 0 & \text{sinon,} \end{cases} \quad (1.44)$$

$$(B_{\mathbf{v}})_{K,L} = \begin{cases} B_{\mathbf{v}}^K & \text{si } L = K, \\ 0 & \text{si } L \neq K, \end{cases} \quad (1.45)$$

$$(M_{\boldsymbol{\sigma}})_{K,L} = \begin{cases} M_{\boldsymbol{\sigma}}^K & \text{si } L = K, \\ 0 & \text{si } L \neq K, \end{cases} \quad (1.46)$$

$$(R_{\mathbf{v}})_{K,L} = \begin{cases} R_{\mathbf{v}}^K + R_{\mathbf{v}}^{K,K} & \text{si } L = K, \\ R_{\mathbf{v}}^{K,L} & \text{si } L \neq K \text{ et } \partial L \cup \partial K \neq \emptyset, \\ 0 & \text{sinon.} \end{cases} \quad (1.47)$$

1.3.3 Discrétisation en temps

Pour terminer la discrétisation du système (1.37), il faut choisir un schéma en temps. Comme discuté précédemment, le choix de la méthode ADER conduit à des schémas dissipatifs. Aussi, nous travaillons dans le code DIVA avec un schéma Leap-Frog.

Le schéma Leap-Frog à deux variables est basé sur un développement de Taylor et consiste à approcher la dérivée à des pas de temps intermédiaires d'une variable à l'autre. Appliqué à une discrétisation par DGM du système de l'élastodynamique au premier ordre, le schéma obtenu est stable sous une condition CFL.

L'intervalle $[0, T]$ est divisé en pas de temps Δt . Soient \mathbf{v}_h^n l'approximation du vecteur \mathbf{v}_h au temps discret $t = n\Delta t$ et $\boldsymbol{\sigma}_h^{n+1/2}$ l'approximation du vecteur $\boldsymbol{\sigma}_h$ au temps discret

intermédiaire $t = (n + \frac{1}{2})\Delta t$. Le système (1.37) se discrétise alors complètement :

$$\begin{cases} M_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^n}{\Delta t} + R_{\sigma} \sigma_h^{n+1/2} + B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} = 0 \\ M_{\sigma} \frac{\sigma_h^{n+3/2} - \sigma_h^{n+1/2}}{\Delta t} + R_{\mathbf{v}} \mathbf{v}_h^{n+1} = 0 \end{cases} \quad (1.48)$$

Comme nous l'avons expliqué précédemment, les matrices de masses et la matrice de bord sont diagonales par blocs et peuvent donc être inversées facilement. Ainsi, le système (1.48) se réécrit de façon quasi-explicite :

$$\begin{cases} \mathbf{v}_h^{n+1} = (\frac{1}{\Delta t} M_{\mathbf{v}} + \frac{1}{2} B_{\mathbf{v}})^{-1} (\frac{1}{\Delta t} M_{\mathbf{v}} - \frac{1}{2} B_{\mathbf{v}}) \mathbf{v}_h^n - (\frac{1}{\Delta t} M_{\mathbf{v}} + \frac{1}{2} B_{\mathbf{v}})^{-1} R_{\sigma} \sigma_h^{n+1/2} \\ \sigma_h^{n+3/2} = \sigma_h^{n+1/2} - \Delta t M_{\sigma}^{-1} R_{\mathbf{v}} \mathbf{v}_h^{n+1} \end{cases} \quad (1.49)$$

De plus, les calculs sont effectués élément par élément sur les blocs de matrice concernés. Au demi pas de temps courant, les calculs locaux ne dépendent que de valeurs précalculées, permettant une résolution entièrement parallèle, favorisant le contexte HPC.

1.4 État de l'art sur la simulation d'ondes anisotropes pour des phénomènes géophysiques

Dans ce chapitre, nous avons introduit le système de l'élastodynamique et décrit comment il évolue en fonction du matériau dans lequel les ondes élastiques se propagent. Nous avons notamment observé que dans des matériaux anisotropes de type TTI, les ondes se propagent comme dans des matériaux anisotropes de type VTI ayant subi une inclinaison. Comme nous l'avons déjà expliqué en introduction, les travaux décrits dans ce manuscrit sont destinés à être intégrés dans un propagateur élastique utilisé pour l'imagerie profondeur par RTM. Il est évident que la qualité des images dépend de la modélisation du sous-sol qui est faite *via* le propagateur. Pour une modélisation isotrope du sous-sol, préférer l'équation des ondes acoustiques à l'équation des ondes élastiques qui est largement plus coûteuse à résoudre, reste efficace. Cela conduit à une RTM dite acoustique et les images ne sont donc construites que sur la propagation des ondes P dans le sous-sol. Cependant, le sous-sol n'est pas isotrope et une RTM acoustique basée sur cette approximation n'est pas capable de détecter toutes les surfaces géologiques, comme cela est illustré par exemple dans [HZZY09, JJR10, GDDW11] sur le modèle synthétique BP, dont les données sont rappelées Fig. 1.11 et les différences observables entre une RTM isotrope et TTI sont présentées Fig. 1.12.

L'anisotropie du sous-sol est considérée polaire, de type TI, c'est-à-dire avec un axe de symétrie d'orthotropie. Concrètement, cela se traduit par des termes couplés entre les ondes P et les ondes S (plus précisément la composante verticale SV), ce qui proscrit la RTM acoustique. En effet, l'analyse asymptotique de l'équation de Christoffel conduit à une relation de dispersion du 4^{ème} ordre mettant en évidence ce couplage. Pour le cas TTI 2D, cette équation s'écrit :

$$\begin{cases} \omega^4 - [(v_{px}^2 + v_{sz}^2)(\tilde{k}_x^2 + \tilde{k}_y^2) + (v_{pz}^2 + v_{sz}^2 \tilde{k}_z^2)]\omega^2 + v_{px}^2 v_{sz}^2 (\tilde{k}_x^2 + \tilde{k}_y^2) + v_{pz}^2 v_{sz}^2 \tilde{k}_z^4 \\ - [v_{pz}^2 (v_{pn}^2 - v_{px}^2) - v_{sz}^2 (v_{pn}^2 + v_{pz}^2)](\tilde{k}_x^2 + \tilde{k}_y^2) \tilde{k}_z^2 = 0 \end{cases} \quad (1.50)$$

1.4. État de l'art sur la simulation d'ondes anisotropes pour des phénomènes géophysiques

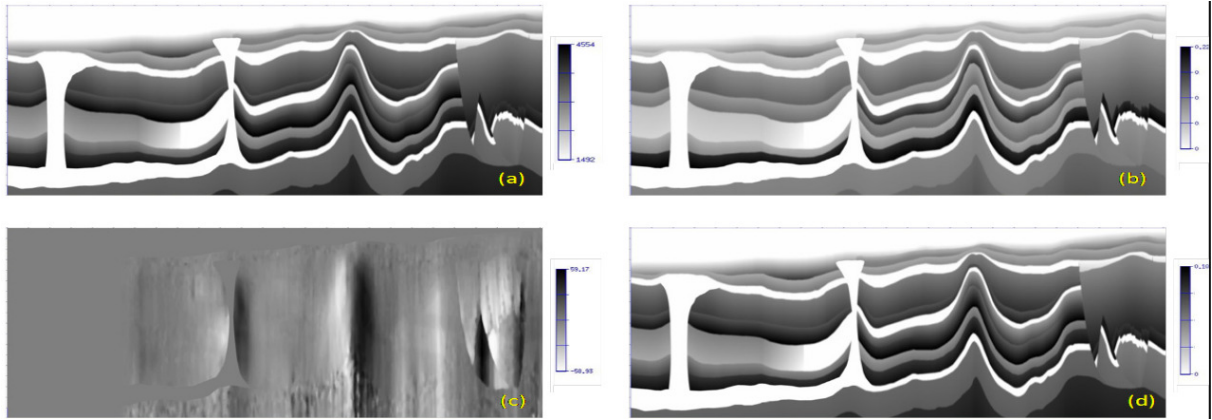


FIGURE 1.11 – Modèles d'anisotropie du cas test synthétique BP, V_p en (a), δ en b, θ en (c) et ϵ en (d) – images issues de [GDDW11]

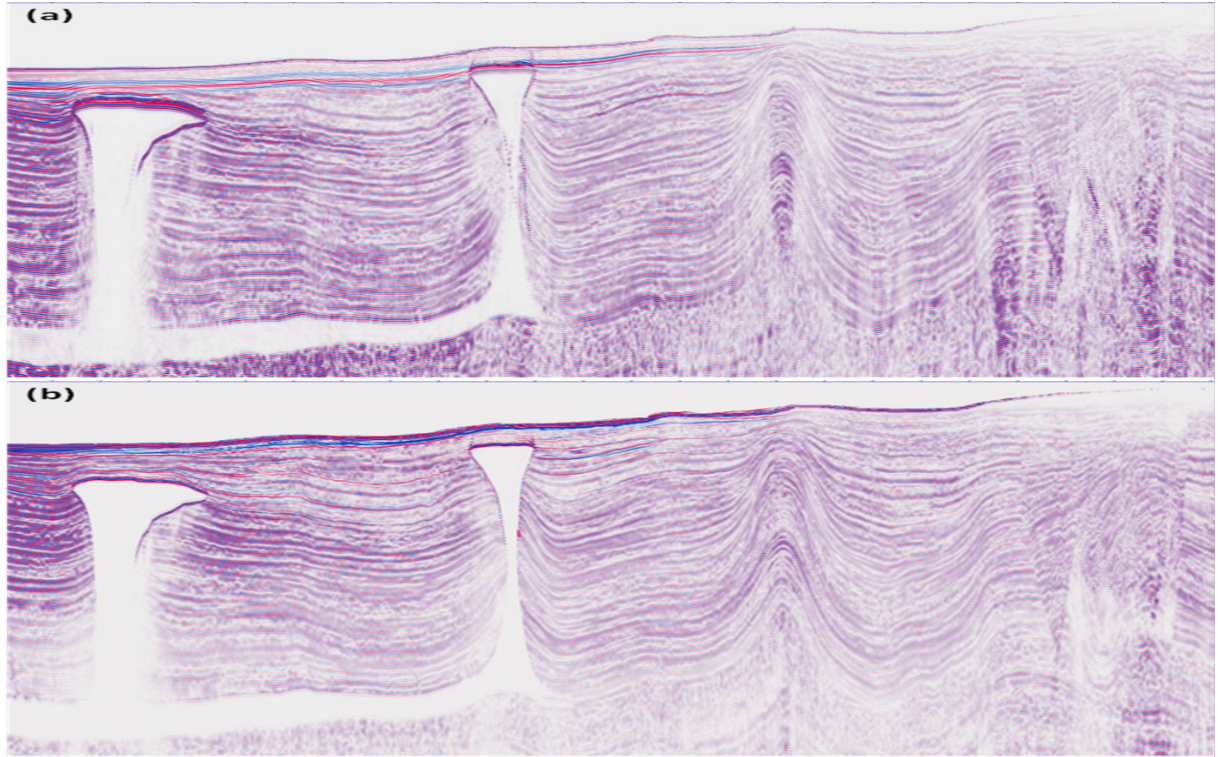


FIGURE 1.12 – Comparaison isotrope/TTI sur le cas test synthétique BP, isotrope en (a), TTI en (b) – images issues de [GDDW11]

où \tilde{k}_x, \tilde{k}_y et \tilde{k}_z sont les nombres d'ondes dans le repère incliné TTI, ω est la fréquence angulaire, v_{px} et v_{pz} sont les deux vitesses des ondes P (selon la direction de propagation dans le repère incliné), v_{sz} est la vitesse des ondes S et v_{pn} correspond à la vitesse NMO¹ des ondes P.

1. NMO (en anglais *normal moveout*), correspond à l'effet de décalage de mesure dû à la séparation entre les sources et les récepteurs.

Des travaux ont alors été consacrés à la RTM basée sur des modèles approchés. Par exemple, Alkhalifah [Alk98] propose une approximation dite “acoustique TI” en reformulant la relation de dispersion TI comme un système de deux équations du deuxième ordre couplées et en fixant la vitesse des ondes S à zéro le long de l’axe de symétrie. Il est alors possible de se focaliser sur les ondes P et de formuler une RTM pseudo-acoustique VTI : [Alk00]. Cependant, fixer la vitesse des ondes S à zéro est un moyen trop radical de simplification qui induit des instabilités importantes, comme le montre [GZR04].

L’idée d’Alkhalifah a ensuite été reprise dans [ZZB06a, DFF08, FDF10] pour améliorer la RTM pseudo-acoustique VTI puis étendue à la RTM pseudo-acoustique TTI dans [ZZB06b, DBL07, ZZ08, LHZAP⁺08, FDF09]. Une autre piste a été suivie par Duveneck et al. [DMBP08] qui ont proposé en 2008 de partir des équations de Hooke (1.3) et du mouvement (1.1) puis de fixer la vitesse de propagation des ondes S pour formuler une RTM pseudo-acoustique VTI. Cette technique a l’avantage de donner un sens physique aux variables des équations du deuxième ordre obtenues. L’idée est alors reprise et étendue par rotation au cas TTI comme cela est fait dans [ZZ09, LBZN09, ZZZ11, DB11].

Les approches d’Alkhalifah et Duveneck et al. sont néanmoins très similaires. Les équations sont toujours couplées et quelques artefacts dus aux ondes S subsistent. Pour parfaire la RTM acoustique, d’autres équipes ont proposé un découplage des équations de dispersion pour ne modéliser que les ondes P : [LMJ⁺09, PUS11] pour le VTI et [ZPS12] pour le TTI. Ces méthodes sont pseudo-spectrales et elles requièrent des implémentations numériques particulières.

Plus récemment, Yan et Sava ont proposé de séparer les conditions d’imagerie des ondes P et des ondes S élastiques pour des milieux isotropes dans [YS08], puis VTI dans [YS09] et TTI dans [YS11], le but fixé étant de l’appliquer à une RTM élastique non-simplifiée. Ce principe de séparation basé sur la décomposition de Helmholtz est connu depuis longtemps pour les équations isotropes, voir [AR02], que Dellinger et Etgen avaient déjà étendu au cas VTI homogène dans [DE90]. En pratique, la RTM élastique est beaucoup plus coûteuse en ressource informatique que la RTM pseudo-acoustique et la reconstruction de l’image dans ce cas est beaucoup plus complexe, voir un exemple en isotrope 2D Fig. 1.13.

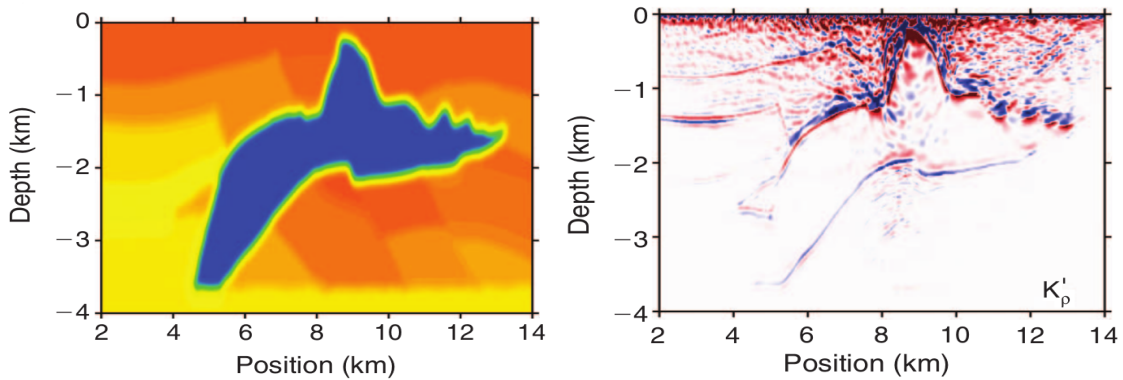


FIGURE 1.13 – Exemple de RTM élastique isotrope 2D, modèle de vitesse (gauche), cartographie des réflecteurs (droite) – images issues de [ZLNM⁺09]

1.4. État de l'art sur la simulation d'ondes anisotropes pour des phénomènes géophysiques

Si son principe est établi depuis bien longtemps, la RTM ne fait qu'évoluer au gré des progrès de la modélisation mathématique et du calcul scientifique sur architectures parallèles. Tout a commencé en considérant le sous-sol isotrope puis VTI puis maintenant TTI, et la prise en compte du rôle des ondes S s'intensifie jusqu'à les intégrer au problème.

Toutes les simulations sont bien évidemment réalisées à l'échelle locale. Il est donc nécessaire de limiter le domaine de calcul par une frontière qui ne doit pas influencer les simulations. La plupart des travaux existants utilisent dans le meilleur des cas des conditions de bord PML, en général celles de [CT01], ou des conditions de Dirichlet ou de surface libre qui sont réfléchissantes. Dans ce cas, il faut un nombre de sources très élevé pour que les réflexions soient atténuées au maximum par effet de sommation dans le processus RTM. Néanmoins, il reste des réflexions parasites dans l'image finale. Des conditions de bord qui génèrent des réflexions minimales sont donc préférables. Dans le cas des PML, c'est en général une couche absorbante isotrope ou VTI qui entoure le domaine d'étude TTI, ce qui génère des réflexions parasites également. Si la couche absorbante est TTI, soit c'est un cas d'étude bien choisi, soit la simulation est en temps très court, car il est démontré dans [BFJ03] que la simulation devient alors rapidement instable. Le recours à des conditions aux limites absorbantes peut alors s'avérer plus intéressant car les faibles réflexions peuvent être diminuées en prenant un nombre suffisant de sources et les problèmes d'instabilité peuvent également être résolus.

Chapitre 2

Conditions aux limites absorbantes pour des matériaux VTI

Ce chapitre se veut être un préambule à la construction de CLA pour des matériaux TTI. Dans ce chapitre, nous revenons sur une méthodologie de construction des CLA qui a été introduite par Engquist et Majda [EM77, EM79] dans les années 70, en suivant la théorie élaborée par Taylor [Tay81] pour suivre la propagation des singularités des solutions de problèmes hyperboliques. Nous nous sommes intéressés de près à cette méthode de construction car elle est particulièrement bien adaptée aux équations hyperboliques du premier ordre. Nous traitons d'abord le cas le plus simple de matériaux isotropes puis nous l'appliquons aux matériaux anisotropes de type VTI. Afin d'alléger l'exposé des calculs, nous détaillons le cas de la dimension deux. La dimension trois sera traitée par la suite pour des matériaux TTI, incluant aussi le cas VTI 3D. Ce chapitre est illustré de résultats numériques isotropes et VTI qui serviront de référence pour apprécier les résultats obtenus pour des matériaux TTI.

2.1 Construction d'une CLA isotrope par diagonalisation

2.1.1 Construction d'une CLA pour un matériau isotrope 2D

Pour simplifier la présentation de la méthode, nous considérons d'abord le cas de la dimension deux. La dimension trois sera traitée par la suite pour des matériaux TTI, incluant aussi les cas isotrope et VTI 3D. Pour conserver les notations classiques de la géophysique, nous nous plaçons donc dans le plan (x, z) . Dans le cas 2D, la matrice d'élasticité (1.18) devient :

$$C_{iso,2D} = \begin{bmatrix} \rho V_p^2 & \rho(V_p^2 - 2V_s^2) & 0 \\ & \rho V_p^2 & 0 \\ & & \rho V_s^2 \end{bmatrix} \quad (2.1)$$

Le système de l'élastodynamique au premier ordre (1.5) s'écrit alors :

$$\begin{cases} \rho \partial_t v_x &= \partial_x \sigma_{xx} + \partial_z \sigma_{xz} \\ \rho \partial_t v_z &= \partial_x \sigma_{xz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= \rho V_p^2 \partial_x v_x + \rho(V_p^2 - 2V_s^2) \partial_z v_z \\ \partial_t \sigma_{zz} &= \rho(V_p^2 - 2V_s^2) \partial_x v_x + \rho V_p^2 \partial_z v_z \\ \partial_t \sigma_{xz} &= \rho V_s^2 (\partial_x v_z + \partial_z v_x) \end{cases} \quad (2.2)$$

et le système au deuxième ordre (1.4) a pour expression :

$$\begin{cases} \rho \partial_t^2 v_x &= \rho V_p^2 \partial_x^2 v_x + \rho(V_p^2 - V_s^2) \partial_x \partial_z v_z + \rho V_s^2 \partial_z^2 v_x \\ \rho \partial_t^2 v_z &= \rho V_s^2 \partial_x^2 v_z + \rho(V_p^2 - V_s^2) \partial_x \partial_z v_x + \rho V_p^2 \partial_z^2 v_z \end{cases} \quad (2.3)$$

Construire une CLA consiste à développer une condition de bord qui approche en un certain sens un opérateur qui décrit la transmission parfaite d'une onde au travers d'une surface donnée. Cet opérateur gouverne une condition aux limites transparente et dans les cas réalistes, il est global en temps et espace, ce qui rend difficile son intégration dans des schémas numériques. Notamment, il rompt la structure creuse des matrices éléments finis et plusieurs écoles ont vu le jour pour la construction d'opérateurs approchés qui préservent l'architecture des matrices discrètes. Certains suivent les idées d'Engquist et Majda [EM77, EM79] qui proposent d'exploiter les propriétés algébriques du système écrit dans le domaine de Fourier après avoir gelé la variable normale liée à la frontière sur laquelle la CLA est à poser. D'autres construisent les conditions en partant de l'expression de la solution analytique du problème, comme dans [BT80, GK95] par exemple. Dans ce cas, les conditions sont écrites pour des surfaces particulières telles que la sphère ou l'ellipsoïde. Nous avons choisi de travailler dans le cadre proposé par Engquist et Majda. Dans notre cas, la frontière est plane, toujours parallèle à un des axes. Dans la suite de ce chapitre, nous allons nous concentrer sur le cas d'une frontière verticale (un des bords du domaine d'étude) pour laquelle la normale est donnée par le vecteur directeur \mathbf{e}_x . Sans perte de généralité, nous allons donc supposer que le bord est donné par la région du plan $\{x = 0\}$ et que le domaine d'intérêt se trouve dans la région $\{x < 0\}$. Nous réécrivons alors le système dans le domaine de Fourier en gelant toute opération dans la variable x . Soient (ω, ξ) les variables de Fourier associées à (t, z) . Le système (2.3) devient :

$$\begin{cases} \rho V_p^2 \partial_x^2 \hat{v}_x - i\xi \rho(V_p^2 - V_s^2) \partial_x \hat{v}_z + (\rho \omega^2 - \rho V_s^2 \xi^2) \hat{v}_x &= 0 \\ \rho V_s^2 \partial_x^2 \hat{v}_z - i\xi \rho(V_p^2 - V_s^2) \partial_x \hat{v}_x + (\rho \omega^2 - \rho V_p^2 \xi^2) \hat{v}_z &= 0 \end{cases} \quad (2.4)$$

Nous obtenons alors une équation différentielle ordinaire en la variable x de la forme :

$$\partial_x W + MW = 0 \quad (2.5)$$

où $W = (\hat{v}_x, \hat{v}_z, \partial_x \hat{v}_x, \partial_x \hat{v}_z)$ et :

$$M = M_{iso,2D} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ \frac{\omega^2 - V_s^2 \xi^2}{V_p^2} & 0 & 0 & \frac{-i\xi(V_p^2 - V_s^2)}{V_p^2} \\ 0 & \frac{\omega^2 - V_p^2 \xi^2}{V_s^2} & \frac{-i\xi(V_p^2 - V_s^2)}{V_s^2} & 0 \end{bmatrix} \quad (2.6)$$

L'équation (2.5) admet une solution de la forme : $W = W_0 e^{Mx}$ dont les composantes $W_{i1 \leq 4}$ s'écrivent :

$$W_i = \sum_{k=1}^4 A_i P_{ik} e^{\Lambda_i x}, \quad \forall i = 1, \dots, 4$$

Les coefficients A_i sont des constantes à déterminer et (Λ_i, P_i) désigne le i -ème couple "valeur propre vecteur propre" de la matrice M définie par (2.6). Chaque vecteur P_i a 4 composantes $P_{ik}, 1 \leq k \leq 4$.

2.1. Construction d'une CLA isotrope par diagonalisation

Dans le cas isotrope, les valeurs propres Λ_i et la matrice des vecteurs propres associés P sont faciles à déterminer, sans avoir recours à un logiciel de calcul formel. Nous obtenons :

$$\Lambda_{iso,2D} = \begin{bmatrix} \lambda_p \\ -\lambda_p \\ \lambda_s \\ -\lambda_s \end{bmatrix} \quad P_{iso,2D} = \begin{bmatrix} \lambda_p & \lambda_p & i\xi & i\xi \\ i\xi & -i\xi & -\lambda_s & \lambda_s \\ -\lambda_p^2 & \lambda_p^2 & -i\xi\lambda_s & i\xi\lambda_s \\ -i\xi\lambda_p & -i\xi\lambda_p & \lambda_s^2 & \lambda_s^2 \end{bmatrix} \quad (2.7)$$

où λ_p et λ_s sont les valeurs propres associées respectivement aux ondes P et aux ondes S, c'est-à-dire :

$$\begin{cases} \lambda_p &= \frac{\sqrt{\xi^2 V_p^2 - \omega^2}}{V_p} \\ \lambda_s &= \frac{\sqrt{\xi^2 V_s^2 - \omega^2}}{V_s} \end{cases} \quad (2.8)$$

Le domaine de propagation étant défini comme le demi-plan $\{x \leq 0\}$, les solutions portées par des exponentielles à exposant négatif sont à éliminer car elles ne sont pas d'énergie finie. Les solutions de l'équation (2.5) s'écrivent alors pour ses deux premières composantes :

$$\begin{cases} \widehat{v}_x &= A\lambda_p e^{\lambda_p x} + Bi\xi e^{\lambda_s x} \\ \widehat{v}_z &= Ai\xi e^{\lambda_p x} - B\lambda_s e^{\lambda_s x} \end{cases} \quad (2.9)$$

et pour ses deux dernières composantes :

$$\begin{cases} \partial_x \widehat{v}_x &= -A\lambda_p^2 e^{\lambda_p x} - Bi\xi\lambda_s e^{\lambda_s x} \\ \partial_x \widehat{v}_z &= -Ai\xi\lambda_p e^{\lambda_p x} + B\lambda_s^2 e^{\lambda_s x} \end{cases} \quad (2.10)$$

En se plaçant sur la frontière absorbante, c'est-à-dire en $x = 0$, la relation (2.9) se simplifie en :

$$\begin{cases} \widehat{v}_x(0) &= A\lambda_p + Bi\xi \\ \widehat{v}_z(0) &= Ai\xi - B\lambda_s \end{cases} \quad (2.11)$$

De même pour le système (2.10) :

$$\begin{cases} \partial_x \widehat{v}_x(0) &= -A\lambda_p^2 - Bi\xi\lambda_s \\ \partial_x \widehat{v}_z(0) &= -Ai\xi\lambda_p + B\lambda_s^2 \end{cases} \quad (2.12)$$

Nous rappelons alors que :

$$\begin{cases} \partial_t \sigma_{xx} &= \rho V_p^2 \partial_x v_x + \rho(V_p^2 - 2V_s^2) \partial_z v_z \\ \partial_t \sigma_{xz} &= \rho V_s^2 (\partial_x v_z + \partial_z v_x) \end{cases} \quad (2.13)$$

qui devient, après transformation de Fourier partielle en t et z :

$$\begin{cases} -i\omega \widehat{\sigma_{xx}} &= \rho V_p^2 \partial_x \widehat{v}_x - i\xi \rho (V_p^2 - 2V_s^2) \widehat{v}_z \\ -i\omega \widehat{\sigma_{xz}} &= \rho V_s^2 (\partial_x \widehat{v}_z - i\xi \widehat{v}_x) \end{cases} \quad (2.14)$$

Il est alors possible d'injecter ces relations dans le système (2.12) pour obtenir :

$$\begin{cases} -i\omega\widehat{\sigma_{xx}} &= \rho V_p^2[-A\lambda_p^2 - Bi\xi\lambda_s] - i\xi\rho(V_p^2 - 2V_s^2)\widehat{v}_z \\ -i\omega\widehat{\sigma_{xz}} &= \rho V_s^2[-Ai\xi\lambda_p + B\lambda_s^2] - i\xi\rho V_s^2\widehat{v}_x \end{cases} \quad (2.15)$$

Il reste maintenant à fixer les constantes A et B en fonction de $\widehat{\mathbf{v}}$ et pour cela, nous utilisons les équations (2.11) que nous injectons dans le système (2.15).

Il vient

$$\begin{cases} A &= \frac{1}{\xi^2 - \lambda_p\lambda_s}(-\lambda_s\widehat{v}_x - i\xi\widehat{v}_z) \\ B &= \frac{1}{\xi^2 - \lambda_p\lambda_s}(-i\xi\widehat{v}_x + \lambda_p\widehat{v}_z) \end{cases} \quad (2.16)$$

Remarque 2.1 Cette méthode, si elle est appliquée directement comme proposé par Engquist et Majda, peut s'avérer extrêmement technique pour la construction de CLA d'ordre élevé, comme cela est illustré par exemple dans [Hal80]. Toutefois, l'introduction de variables auxiliaires peut la rendre plus pratique comme par exemple dans [Tso99]. Nous n'aborderons pas cette question dans ce manuscrit car nous visons une application qui se contente a priori de CLA d'ordre peu élevé. Le caractère novateur de notre travail sera donc d'écrire une condition stable pour des matériaux anisotropes TTI à des fins d'utilisation dans un code de RTM.

La relation (2.15) écrite avec les constantes (2.16) fait intervenir les valeurs propres de la matrice (2.6) qui s'écrivent dans le domaine de Fourier en fonction d'une racine carrée. En général, le retour au domaine spatio-temporel se fait en appliquant une transformée de Fourier inverse, mais, en procédant ainsi, la condition transparente obtenue s'exprimerait comme un opérateur de Fourier intégral en temps et espace. L'intégration de la condition aux limites dans la formulation détruirait donc la structure des matrices qui ont l'avantage d'être creuses, rendant le système implicite en couplant tous les éléments du bord. De plus cela nécessiterait de stocker l'historique de la solution sur le bord.

Cependant, il est possible de localiser le symbole de l'opérateur en appliquant une approximation de Taylor ou de Padé. Dans le premier cas, la racine carrée est remplacée par un polynôme, ce qui donne des opérateurs différentiels dans le domaine spatio-temporel. Dans le second cas, cela mène à des fractions rationnelles qui permettent aussi de manipuler des opérateurs différentiels, quitte à introduire des variables auxiliaires. Il est alors nécessaire d'identifier un petit paramètre et dans notre cas, il s'agit de $\frac{\xi^2}{\omega^2}$ qui est inférieur à 1 dans le cône de propagation et qui est toujours petit quand la fréquence augmente. L'approximation la plus basique de la racine carrée est donnée par la formule :

$$\sqrt{1-x} = 1 + o(x) \quad (2.17)$$

Dans la suite, nous utiliserons le symbole " \approx " pour indiquer l'approximation en $o(x)$ pour x au voisinage de zéro. Nous rappelons qu'ici, x désigne la variable $\frac{\xi^2}{\omega^2}$, qui est proche de zéro dans le cône de propagation.

2.1. Construction d'une CLA isotrope par diagonalisation

Considérons d'abord les valeurs propres définies à l'équation (2.8). Nous avons :

$$\begin{cases} \lambda_p &= -\frac{i\omega}{V_p} \sqrt{1 - \frac{\xi^2 V_p^2}{\omega^2}} \approx -\frac{i\omega}{V_p} \\ \lambda_s &= -\frac{i\omega}{V_s} \sqrt{1 - \frac{\xi^2 V_s^2}{\omega^2}} \approx -\frac{i\omega}{V_s} \end{cases} \quad (2.18)$$

De plus,

$$\frac{1}{\xi^2 - \lambda_p \lambda_s} = \frac{1}{\omega^2} \frac{1}{\frac{\xi^2}{\omega^2} + \frac{1}{V_p V_s} \sqrt{1 - \frac{\xi^2 V_p^2}{\omega^2}} \sqrt{1 - \frac{\xi^2 V_s^2}{\omega^2}}} \approx \frac{V_p V_s}{\omega^2} \quad (2.19)$$

ce qui conduit à l'expression approchée :

$$\begin{cases} A &\approx \frac{V_p V_s}{\omega^2} \left(\frac{i\omega}{V_s} \hat{v}_x - i\xi \hat{v}_z \right) \\ B &\approx \frac{V_p V_s}{\omega^2} \left(-i\xi \hat{v}_x - \frac{i\omega}{V_p} \hat{v}_z \right) \end{cases} \quad (2.20)$$

Il ne reste plus qu'à injecter (2.20) et (2.18) dans (2.15) :

$$\begin{cases} -i\omega \widehat{\sigma_{xx}} &\approx \rho V_p^2 \frac{V_p V_s}{\omega^2} \left[\left(\frac{i\omega}{V_s} \hat{v}_x + i\xi \hat{v}_z \right) \frac{\omega^2}{V_p^2} + \left(-i\xi \hat{v}_x - \frac{i\omega}{V_p} \hat{v}_z \right) i\xi \frac{i\omega}{V_s} \right] - i\xi \rho (V_p^2 - 2V_s^2) \hat{v}_z \\ -i\omega \widehat{\sigma_{xz}} &\approx \rho V_s^2 \frac{V_p V_s}{\omega^2} \left[\left(\frac{i\omega}{V_s} \hat{v}_x - i\xi \hat{v}_z \right) i\xi \frac{i\omega}{V_p} - \left(-i\xi \hat{v}_x + \frac{i\omega}{V_p} \hat{v}_z \right) \frac{\omega^2}{V_s^2} \right] - i\xi \rho V_s^2 \hat{v}_x \end{cases} \quad (2.21)$$

ce qui se réécrit :

$$\begin{cases} \omega^2 \widehat{\sigma_{xx}} &\approx -\rho V_p (\omega^2 + \xi^2 V_p^2) \hat{v}_x + i\xi \rho (V_s V_p - 2V_s^2) \hat{v}_z \\ \omega^2 \widehat{\sigma_{xz}} &\approx -\rho V_s (\omega^2 + \xi^2 V_s^2) \hat{v}_z + i\xi \rho (2V_s^2 - V_s V_p) \hat{v}_x \end{cases} \quad (2.22)$$

En appliquant une transformée de Fourier inverse, il vient finalement la condition aux limites absorbantes :

$$\begin{cases} \partial_t^2 \sigma_{xx} &= -\rho V_p (\partial_t^2 v_x + V_p^2 \partial_z^2 v_x) + \rho (V_s V_p - 2V_s^2) \partial_z v_z \\ \partial_t^2 \sigma_{xz} &= -\rho V_s (\partial_t^2 v_z + V_s^2 \partial_z^2 v_x) + \rho (2V_s^2 - V_s V_p) \partial_z v_x \end{cases} \quad (2.23)$$

Nous observons que la condition obtenue contient des dérivées dans la direction z . Il est assez courant d'éliminer ces termes (voir [Rey78] et [Hal80]) en s'appuyant que le fait que les ondes de plus fortes amplitudes sont concentrées dans un cône dont l'axe de révolution est selon la direction de la normale. Dans notre cas, nous pouvons donc supposer que les ondes réfléchies de forte amplitude ne dépendent pas de z . Nous proposons alors d'utiliser comme CLA, la condition aux limites plus simple, obtenue en éliminant les termes comportant des dérivées en z et en intégrant en temps :

$$\begin{cases} \sigma_{xx} &= -\rho V_p v_x \\ \sigma_{xz} &= -\rho V_s v_z \end{cases} \quad (2.24)$$

2.1.2 Étude expérimentale pour l'élastique isotrope 2D

Cas tests

Comme nous l'avons dit à plusieurs reprises, notre objectif est d'améliorer un logiciel d'imagerie du sous-sol en permettant de considérer des matériaux anisotropes. Dans cette première partie de la thèse, nous nous focalisons sur la construction de conditions aux limites absorbantes dont la validation requiert de quantifier l'amplitude des ondes réfléchies par la surface absorbante. Dans un cas réaliste, les portions du sous-sol sont constituées de nombreuses couches géologiques différentes. Le phénomène est donc décrit par un nombre de réflexions très important et il n'est pas évident de discriminer les réflexions réelles internes des réflexions parasites générées par la frontière du domaine d'étude. Nous avons donc commencé par considérer des cas test simples, à savoir un cas homogène, c'est-à-dire constitué d'un seul matériau, puis des cas bi-couches, de type bande ou coin, voir Fig. 2.1, de dimension 10 km² dont les caractéristiques physiques sont décrites dans le tableau 2.1 (où nous considérons que la masse volumique en 2D est une masse surfacique).

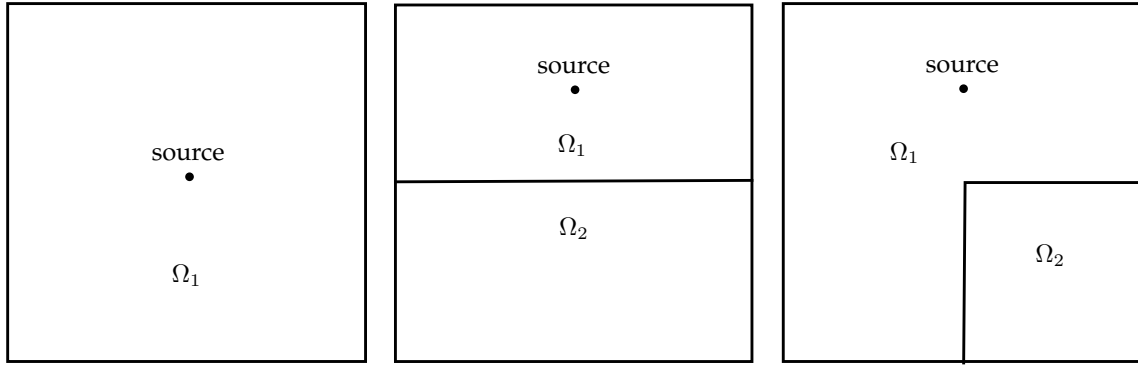


FIGURE 2.1 – Cas tests isotropes 2D homogène (gauche), bi-couches (centre) et coin (droite)

milieu	ρ [kg.m ⁻²]	V_p [m.s ⁻¹]	V_s [m.s ⁻¹]
Ω_1	1	3000	1800
Ω_2	1	2200	1250

TABLE 2.1 – Caractéristiques physiques des cas-tests isotropes

Source et condition initiale

Nous allons suivre la procédure décrite dans la section 1.1.2 pour construire une condition initiale et une source afin de ne générer qu'une onde P, qu'une onde S ou la combinaison d'une onde P et d'une onde S.

En élastique isotrope 2D, la matrice de l'équation de dispersion s'écrit :

$$E_{iso,2D} = \begin{pmatrix} \rho V_p^2 k_x^2 + \rho V_s^2 k_z^2 & \rho(V_p^2 - V_s^2) k_x k_z \\ \rho(V_p^2 - V_s^2) k_x k_z & \rho V_s^2 k_x^2 + \rho V_p^2 k_z^2 \end{pmatrix} \quad (2.25)$$

2.1. Construction d'une CLA isotrope par diagonalisation

Les vecteurs propres de (2.25) sont orthogonaux et font intervenir les symboles de Fourier :

$$\Lambda_p = \begin{pmatrix} ik_x \\ ik_z \end{pmatrix}, \quad \Lambda_s = \begin{pmatrix} -ik_z \\ ik_x \end{pmatrix} \quad (2.26)$$

ce qui conduit aux opérateurs :

$$B_p = \begin{pmatrix} \partial_x \\ \partial_z \end{pmatrix}, \quad B_s = \begin{pmatrix} -\partial_z \\ \partial_x \end{pmatrix} \quad (2.27)$$

Nous rappelons que notre choix de formule pour les conditions initiales est $\underline{\sigma}_0 = 0$ et $\mathbf{v}_0(\mathbf{x}) = B_\star(e^{-\pi^2\|\mathbf{x}-\mathbf{x}_0\|^2})$. Pour l'équation élastique isotrope 2D, cela conduit aux différents choix :

$$\begin{cases} \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} (x - x_0) \\ (z - z_0) \end{pmatrix} e^{-\pi^2\|\mathbf{x}-\mathbf{x}_0\|^2}, & \text{pour ne générer qu'une onde P,} \\ \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} -(z - z_0) \\ (x - x_0) \end{pmatrix} e^{-\pi^2\|\mathbf{x}-\mathbf{x}_0\|^2}, & \text{pour ne générer qu'une onde S,} \end{cases} \quad (2.28)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Pour utiliser une source ponctuelle à la place d'une condition initiale, nous rappelons que sa forme lorsqu'elle est appliquée sur l'équation du tenseur des contraintes est $s_R(t)\underline{s}(\mathbf{x})$, où $s_R(t)$ est une amplitude (en l'occurrence une ondelette de Ricker définie en (1.9)). Il suffit ensuite de chercher \underline{s} telle que $\nabla \cdot \underline{s} = B_\star(\delta_{\mathbf{x}_0})$. Ecrivons cette égalité pour B_p :

$$\begin{pmatrix} \partial_x s_{xx} + \partial_z s_{xz} \\ \partial_x s_{zx} + \partial_z s_{zz} \end{pmatrix} = \begin{pmatrix} \partial_x \\ \partial_z \end{pmatrix} \delta_{\mathbf{x}_0} \quad (2.29)$$

L'identification terme à terme conduit à une solution simple : $s_{xz} = s_{zx} = 0$ et $s_{xx} = s_{zz} = \delta_{\mathbf{x}_0}$. En procédant de la même façon pour B_s , les différents choix de sources pour l'équation élastique isotrope 2D s'écrivent finalement :

$$\begin{cases} \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \text{pour ne générer qu'une onde P,} \\ \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, & \text{pour ne générer qu'une onde S,} \end{cases} \quad (2.30)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Remarque 2.2 Nous retrouvons la source pseudo-acoustique 2D en onde P (1.8) utilisée initialement dans le code DIVA isotrope.

Résultats numériques

Sur les trois configurations de domaine décrites Fig. 2.1 et dans le tableau 2.1, nous allons considérer une source élastique (ondes P et S) puis des sources pseudo-acoustiques (onde P ou onde S). Les résultats sur le domaine homogène sont présentés Fig. 2.2, ceux des domaines bi-couches bande et coin sont présentés Fig. 2.3 et 2.4 respectivement. Le module du vecteur vitesse est représenté pour différents temps de simulation. À $t = 1s$, nous observons le front d'onde, ce qui permet de figer l'échelle des représentations. À $t = 2s$ ou $t = 3s$, nous commençons à observer le travail des CLA car le front d'onde a atteint les bords du domaine de calcul. Nous laissons ensuite les calculs continuer jusqu'à $t = 5s$ puis $t = 7s$. Nous observons ainsi les réflexions parasites propres aux bords absorbants. Les mêmes résultats sont ensuite présentés à l'échelle de couleurs des réflexions, calculée au temps, en fonction des cas, $t = 5s$ (homogène) ou $t = 7s$ (bi-couches). Toutes les valeurs des échelles sont regroupées dans le tableau 2.2.

Nous observons un comportement connu : plus l'angle d'incidence de l'onde s'éloigne de l'axe normal, plus les réflexions sont fortes, avec une absorption totale à incidence normale. C'est tout à fait cohérent avec le mode d'approximation qui a été choisi pour localiser la condition aux limites transparente. De plus, les ondes S génèrent des réflexions plus fortes que celles des ondes P, ce qui justifie aussi le choix de considérer une source en onde P seule, pour mieux étudier les réflexions et se rapprocher des modèles pseudo-acoustiques utilisés par les géophysiciens.

source	temps	homogène	bi-couche bande	bi-couche coin
PS	1s / 1s / 1s	1.56	1.61	1.60
	5s / 7s / 7s	0.11	0.27	0.26
P	1s / 1s / 1s	1.22	1.45	1.55
	5s / 7s / 7s	0.04	0.14	0.10
S	1s / 1s / 1s	1.57	1.57	1.57
	5s / 7s / 7s	0.11	0.27	0.26

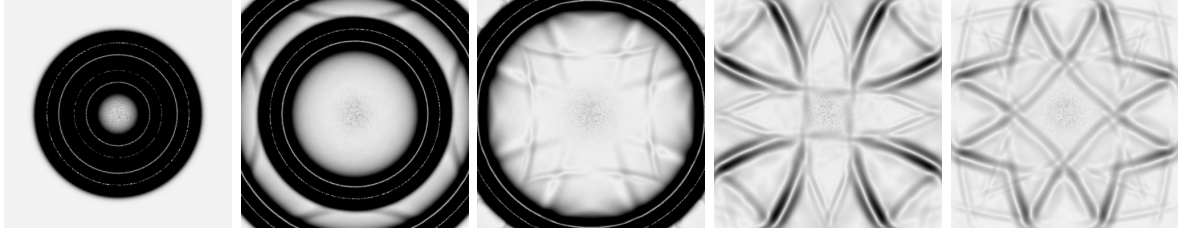
TABLE 2.2 – Valeurs maximales du module du vecteur vitesse pour l'étude isotrope 2D

Remarque 2.3 *Les résultats que nous obtenons sont en adéquation avec ceux des autres codes développés dans l'équipe. Il était cependant nécessaire de les lancer pour deux raisons : ils permettent de valider la formulation isotrope que nous utilisons et ils seront utiles par la suite en nous fournissant des cas de référence en vue de la validation des cas d'anisotropie.*

2.1. Construction d'une CLA isotrope par diagonalisation



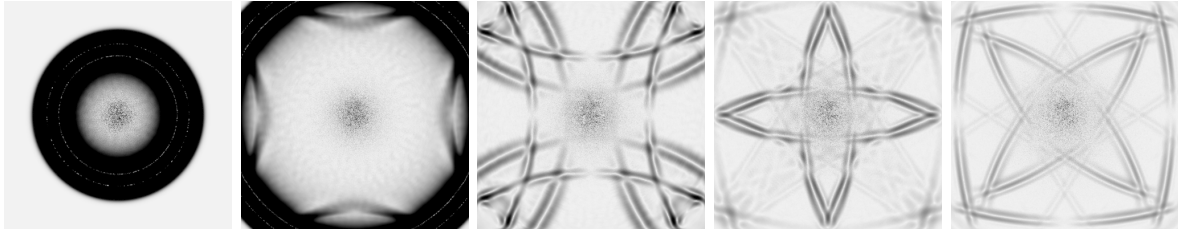
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$



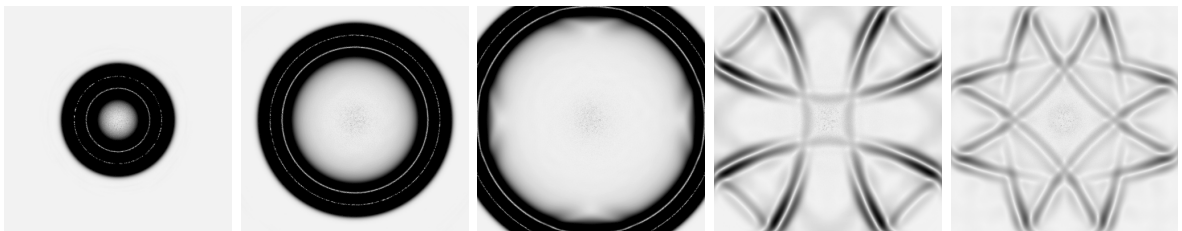
(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 5s$



(e) source S, échelle de couleur fixée à $t = 1s$



(f) source S, échelle de couleur fixée à $t = 5s$

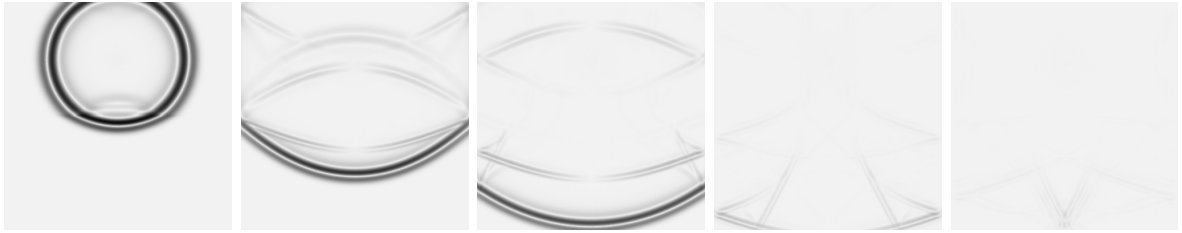
FIGURE 2.2 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration homogène isotrope 2D



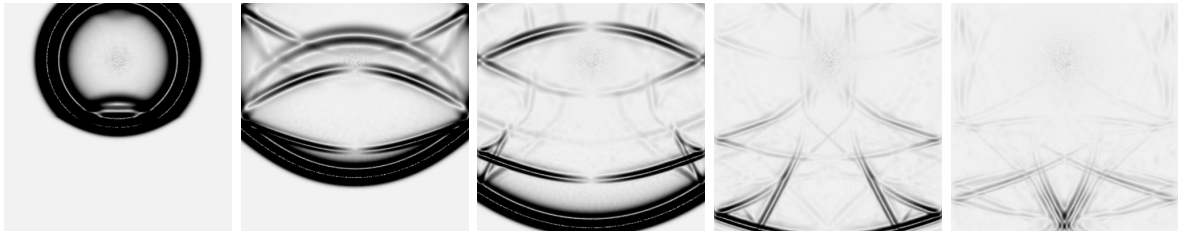
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 7s$



(e) source S, échelle de couleur fixée à $t = 1s$

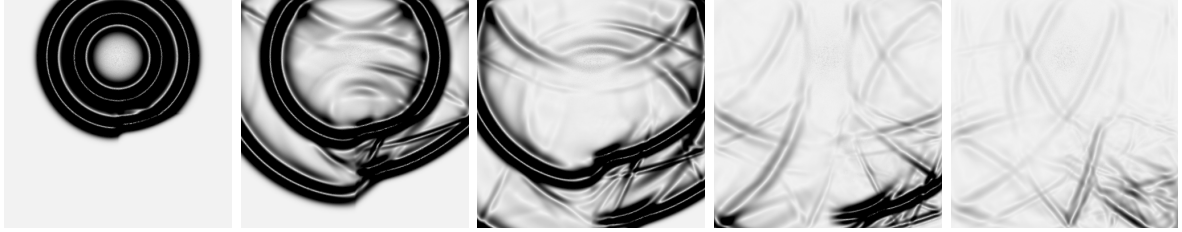


(f) source S, échelle de couleur fixée à $t = 7s$

2.1. Construction d'une CLA isotrope par diagonalisation



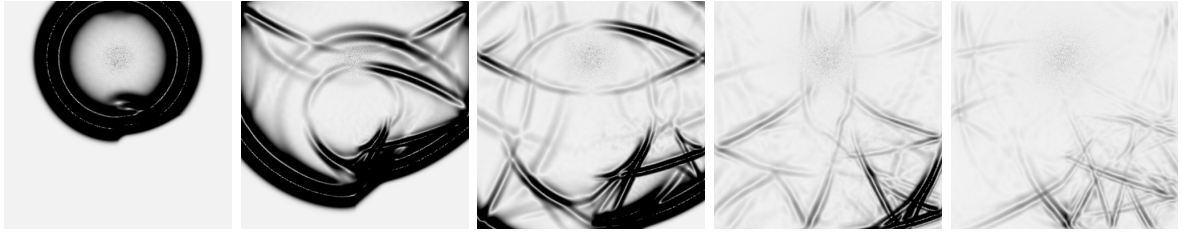
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 7s$



(e) source S, échelle de couleur fixée à $t = 1s$



(f) source S, échelle de couleur fixée à $t = 7s$

FIGURE 2.4 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration bi-couche coin isotrope 2D

2.2 Construction d'une CLA VTI par diagonalisation

2.2.1 Construction d'une CLA pour un matériau VTI 2D

L'anisotropie de type VTI est représentée par des tenseurs dont les coefficients ne sont pas très éloignés des coefficients des tenseurs isotropes. Il nous est ainsi possible d'appliquer le même processus de construction qu'à la section 2.1. Comme auparavant, nous nous focalisons sur le cas 2D car il est plus simple à exposer.

Dans un matériau anisotrope VTI, la matrice d'élasticité (1.20) s'écrit :

$$C_{VTI,2D} = \begin{bmatrix} C_{11} & C_{13} & 0 \\ & C_{33} & 0 \\ & & C_{44} \end{bmatrix} \quad (2.31)$$

Le système de l'élastodynamique au premier ordre (1.5) est alors donné par :

$$\begin{cases} \rho \partial_t v_x &= \partial_x \sigma_{xx} + \partial_z \sigma_{xz} \\ \rho \partial_t v_z &= \partial_x \sigma_{xz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= C_{11} \partial_x v_x + C_{13} \partial_z v_z \\ \partial_t \sigma_{zz} &= C_{13} \partial_x v_x + C_{33} \partial_z v_z \\ \partial_t \sigma_{xz} &= C_{44} (\partial_x v_z + \partial_z v_x) \end{cases} \quad (2.32)$$

et le système au deuxième ordre (1.4) est défini par :

$$\begin{cases} \rho \partial_t^2 v_x &= C_{11} \partial_x^2 v_x + (C_{13} + C_{44}) \partial_x \partial_z v_z + C_{44} \partial_z^2 v_x \\ \rho \partial_t^2 v_z &= C_{44} \partial_x^2 v_z + (C_{13} + C_{44}) \partial_x \partial_z v_x + C_{33} \partial_z^2 v_z \end{cases} \quad (2.33)$$

Comme pour le cas isotrope, nous supposons que la frontière absorbante est paramétrée par $x = 0$. En supposant que le domaine d'intérêt est défini par $x \leq 0$, la normale unitaire extérieure est définie par le vecteur de base \mathbf{e}_x . Soient (ω, ξ) les variables de Fourier associées à (t, z) . Dans le domaine de Fourier, le système (2.33) se réécrit :

$$\begin{cases} C_{11} \partial_x^2 \hat{v}_x - i\xi (C_{13} + C_{44}) \partial_x \hat{v}_z + (\rho\omega^2 - C_{44}\xi^2) \hat{v}_x &= 0 \\ C_{44} \partial_x^2 \hat{v}_z - i\xi (C_{13} + C_{44}) \partial_x \hat{v}_x + (\rho\omega^2 - C_{33}\xi^2) \hat{v}_z &= 0 \end{cases} \quad (2.34)$$

L'équation (2.34) est une équation différentielle ordinaire en la variable x de la forme :

$$\partial_x W + MW = 0 \quad (2.35)$$

avec $W = (\hat{v}_x, \hat{v}_z, \partial_x \hat{v}_x, \partial_x \hat{v}_z)$ et :

$$M = M_{VTI,2D} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ \frac{\rho\omega^2 - C_{44}\xi^2}{C_{11}} & 0 & 0 & \frac{-i\xi(C_{13} + C_{44})}{C_{11}} \\ 0 & \frac{\rho\omega^2 - C_{33}\xi^2}{C_{44}} & \frac{-i\xi(C_{13} + C_{44})}{C_{44}} & 0 \end{bmatrix} \quad (2.36)$$

L'équation (2.35) admet des solutions de la forme : $W = W_0 e^{Mx}$ ie $W_i = \sum_{k=1}^4 A_i P_{ik} e^{\Lambda_i x}$, $\forall i = 1, \dots, 4$, où A_i sont des constantes à déterminer et (Λ_i, P_i) désigne le i -ème couple "valeur

2.2. Construction d'une CLA VTI par diagonalisation

propre vecteur propre" de la matrice M définie en (2.36).

Les valeurs propres Λ et la matrice des vecteurs propres associés P sont plus difficiles à calculer que dans le cas isotrope au sens où il n'est pas aisé de procéder à des calculs à la main. En utilisant le logiciel MapleTM [Maple] (voir annexe A.2), nous obtenons :

$$\Lambda_{VTI,2D} = \begin{bmatrix} \lambda_p \\ -\lambda_p \\ \lambda_s \\ -\lambda_s \end{bmatrix} \quad P_{VTI,2D} = \begin{bmatrix} \lambda_p X_p & \lambda_p X_p & i\xi X_s & i\xi X_s \\ i\xi & -i\xi & -\lambda_s & \lambda_s \\ -\lambda_p^2 X_p & \lambda_p^2 X_p & -i\xi \lambda_s X_s & i\xi \lambda_s X_s \\ -i\xi \lambda_p & -i\xi \lambda_p & \lambda_s^2 & \lambda_s^2 \end{bmatrix} \quad (2.37)$$

avec

$$\begin{cases} X_p = \frac{\xi^2(C_{13} + C_{44})(\rho\omega^2 - \xi^2 C_{33})}{C_{11}C_{44}\lambda_p^2 \left(\frac{\rho\omega^2 - \xi^2 C_{33}}{C_{44}} + \lambda_s^2 \right)} \\ X_s = \frac{(C_{13} + C_{44})(\rho\omega^2 - \xi^2 C_{33})}{C_{11}C_{44} \left(\frac{\rho\omega^2 - \xi^2 C_{33}}{C_{44}} + \lambda_p^2 \right)} \end{cases} \quad (2.38)$$

Les valeurs propres λ_p et λ_s , associées respectivement aux ondes P et aux ondes S sont définies par :

$$\begin{cases} \lambda_p = \sqrt{\alpha\xi^2 + \beta\rho\omega^2 + \sqrt{\gamma\xi^4 + \eta\rho\xi^2\omega^2 + \nu\rho^2\omega^4}} \\ \lambda_s = \sqrt{\alpha\xi^2 + \beta\rho\omega^2 - \sqrt{\gamma\xi^4 + \eta\rho\xi^2\omega^2 + \nu\rho^2\omega^4}} \end{cases} \quad (2.39)$$

avec

$$\begin{cases} \alpha = -\frac{C_{13}^2 - C_{33}C_{11} + 2C_{13}C_{44}}{2C_{11}C_{44}} \\ \beta = -\frac{C_{11} + C_{44}}{2C_{11}C_{44}} \\ \gamma = \alpha^2 - \frac{C_{33}}{C_{11}} \\ \eta = 2\alpha\beta + \frac{C_{33} + C_{44}}{C_{11}C_{44}} \\ \nu = \left(\frac{C_{11} - C_{44}}{2C_{11}C_{44}} \right)^2 \end{cases} \quad (2.40)$$

Remarque 2.4 À notre connaissance, il n'existe qu'un article [Pod12] dans lequel la composée des deux racines carrées est approchée par une formule plus élaborée mais l'approximation n'est valide que dans un cas particulier de symétrie axiale. Dans ce cas, l'opérateur est localisé en espace en utilisant des approximations de Padé et le noyau en temps est approché par des exponentielles.

Remarque 2.5 Les valeurs propres s'expriment ici à l'aide de la composition de deux racines carrées comme illustré dans (2.39). Ce terme compliqué est un témoin du couplage entre les ondes P et les ondes S qui est induit par l'anisotropie. Si les paramètres (1.22) sont fixés à zéro, les valeurs propres se simplifient et cela mène à la forme simple des valeurs propres obtenues dans le cas isotrope (2.8).

Le domaine de propagation étant défini comme le demi-plan $\{x \leq 0\}$, les solutions portées par des exponentielles à exposant négatif sont à éliminer car elles ne sont pas d'énergie finie. Les solutions de l'équation (2.35) s'écrivent alors pour ses deux premières composantes :

$$\begin{cases} \widehat{v}_x &= AX_p \lambda_p e^{\lambda_p x} + BX_s i \xi e^{\lambda_s x} \\ \widehat{v}_z &= Ai \xi e^{\lambda_p x} - B \lambda_s e^{\lambda_s x} \end{cases} \quad (2.41)$$

et pour ses deux dernières composantes :

$$\begin{cases} \partial_x \widehat{v}_x &= -AX_p \lambda_p^2 e^{\lambda_p x} - BX_s i \xi \lambda_s e^{\lambda_s x} \\ \partial_x \widehat{v}_z &= -Ai \xi \lambda_p e^{\lambda_p x} + B \lambda_s^2 e^{\lambda_s x} \end{cases} \quad (2.42)$$

En se plaçant sur la frontière absorbante, c'est-à-dire en $x = 0$, la relation (2.41) se simplifie en :

$$\begin{cases} \widehat{v}_x(0) &= AX_p \lambda_p + BX_s i \xi \\ \widehat{v}_z(0) &= Ai \xi - B \lambda_s \end{cases} \quad (2.43)$$

De même, le système (2.42) devient en $x = 0$:

$$\begin{cases} \partial_x \widehat{v}_x(0) &= -AX_p \lambda_p^2 - BX_s i \xi \lambda_s \\ \partial_x \widehat{v}_z(0) &= -Ai \xi \lambda_p + B \lambda_s^2 \end{cases} \quad (2.44)$$

Nous rappelons maintenant que :

$$\begin{cases} \partial_t \sigma_{xx} &= C_{11} \partial_x v_x + C_{13} \partial_z v_z \\ \partial_t \sigma_{xz} &= C_{44} (\partial_x v_z + \partial_z v_x) \end{cases} \quad (2.45)$$

qui devient, après transformation de Fourier partielle en t et z :

$$\begin{cases} -i\omega \widehat{\sigma_{xx}} &= C_{11} \partial_x \widehat{v}_x - i\xi C_{13} \widehat{v}_z \\ -i\omega \widehat{\sigma_{xz}} &= C_{44} (\partial_x \widehat{v}_z - i\xi \widehat{v}_x) \end{cases} \quad (2.46)$$

Il est alors possible d'injecter ces relations dans le système (2.44) pour obtenir :

$$\begin{cases} -i\omega \widehat{\sigma_{xx}} &= C_{11} [-AX_p \lambda_p^2 - BX_s i \xi \lambda_s] - i\xi C_{13} \widehat{v}_z \\ -i\omega \widehat{\sigma_{xz}} &= C_{44} [-Ai \xi \lambda_p + B \lambda_s^2] - i\xi C_{44} \widehat{v}_x \end{cases} \quad (2.47)$$

Il reste maintenant à fixer les constantes A et B en fonction de $\widehat{\mathbf{v}}$ et pour cela, nous utilisons les équations (2.43).

2.2. Construction d'une CLA VTI par diagonalisation

Il vient :

$$\begin{cases} A = \frac{1}{X_s \xi^2 - X_p \lambda_p \lambda_s} (-\lambda_s \widehat{v}_x - i \xi X_s \widehat{v}_z) \\ B = \frac{1}{X_s \xi^2 - X_p \lambda_p \lambda_s} (-i \xi \widehat{v}_x + X_p \lambda_p \widehat{v}_z) \end{cases} \quad (2.48)$$

Comme pour le cas isotrope, nous localisons la condition en appliquant une approximation de Taylor d'ordre zéro en adoptant la notation “ \approx ” introduite à la section précédente. Le petit paramètre est toujours $\frac{\xi^2}{\omega^2}$ et les valeurs propres définies à l'équation (2.39) se simplifient alors à l'ordre zéro avec notamment la formule de la racine carrée (2.17) :

$$\begin{cases} \lambda_p = -i\omega \sqrt{-\alpha \frac{\xi^2}{\omega^2} - \beta\rho - \sqrt{\gamma \frac{\xi^4}{\omega^4} + \eta\rho \frac{\xi^2}{\omega^2} + \nu\rho^2}} \approx -i\omega \sqrt{-\beta\rho - \sqrt{\nu\rho^2}} = -\sqrt{\rho} \frac{i\omega}{\sqrt{C_{11}}} \\ \lambda_s = -i\omega \sqrt{-\alpha \frac{\xi^2}{\omega^2} - \beta\rho + \sqrt{\gamma \frac{\xi^4}{\omega^4} + \eta\rho \frac{\xi^2}{\omega^2} + \nu\rho^2}} \approx -i\omega \sqrt{-\beta\rho + \sqrt{\nu\rho^2}} = -\sqrt{\rho} \frac{i\omega}{\sqrt{C_{44}}} \end{cases} \quad (2.49)$$

La simplification au premier ordre du système (2.47) est plus complexe que pour le cas isotrope puisqu'elle fait intervenir les termes X_p et X_s . Ce calcul a donc été effectué à l'aide de MapleTM, le détail est donné en annexe A.3. Cela conduit au système :

$$\begin{cases} -i\omega \widehat{\sigma}_{xx} \approx \sqrt{\rho} i\omega \sqrt{C_{11}} \widehat{v}_x + \left(i\xi (C_{11} - \sqrt{C_{11}} \sqrt{C_{44}}) \frac{C_{13} + C_{44}}{C_{11} - C_{44}} - i\xi C_{13} \right) \widehat{v}_z \\ -i\omega \widehat{\sigma}_{xz} \approx -i\xi C_{44} \widehat{v}_x + \sqrt{\rho} i\omega \sqrt{C_{44}} \widehat{v}_z \end{cases} \quad (2.50)$$

En appliquant une transformée de Fourier inverse, il vient finalement la condition aux limites absorbantes :

$$\begin{cases} \partial_t \sigma_{xx} = -\sqrt{\rho} \partial_t \sqrt{C_{11}} v_x + \left(-\partial_z (C_{11} - \sqrt{C_{11}} \sqrt{C_{44}}) \frac{C_{13} + C_{44}}{C_{11} - C_{44}} + \partial_z C_{13} \right) v_z \\ \partial_t \sigma_{xz} = \partial_z C_{44} v_x - \sqrt{\rho} \partial_t \sqrt{C_{44}} v_z \end{cases} \quad (2.51)$$

Comme dans le cas isotrope, nous proposons d'utiliser comme CLA, la condition aux limites plus simple, obtenue en négligeant les dérivées en z (voir [Rey78] et [Hal80]) et en intégrant en temps :

$$\begin{cases} \sigma_{xx} = -\sqrt{\rho C_{11}} v_x \\ \sigma_{xz} = -\sqrt{\rho C_{44}} v_z \end{cases} \quad (2.52)$$

Remarque 2.6 La condition aux limites absorbantes (2.52) avec les coefficients du tenseur élastique est valable pour tous les problèmes orthotropes 2D et donc en particulier pour le VTI. Cette CLA orthotrope 2D d'ordre faible a déjà été formulée dans [Bec91], à l'aide des équations de Christoffel.

En utilisant la valeur des coefficients TI (1.21), nous obtenons finalement la CLA VTI :

$$\begin{cases} \sigma_{xx} = -\rho V_p \sqrt{1 + 2\varepsilon} v_x \\ \sigma_{xz} = -\rho V_s v_z \end{cases} \quad (2.53)$$

Remarque 2.7 Lorsque $\varepsilon = 0$, la CLA VTI (2.53) coïncide avec la CLA isotrope (2.24). De plus, il est à noter que nous avons obtenu une CLA VTI qui ne fait pas intervenir le coefficient de Thomsen δ . Nous reviendrons sur cette propriété plus loin car elle sera cruciale pour mettre en œuvre notre technique de construction d'une CLA TTI.

2.2.2 Construction de CLA VTI 2D par découplage des ondes

Le système de l'élastodynamique portant sur les ondes P et S peut être transformé en deux sous-problèmes pseudo-acoustiques en fixant V_s ou V_p à zéro. Cela n'a pas de sens physique mais permet d'isoler les termes des équations qui n'interviennent que sur l'une des deux ondes. Le même processus de construction de CLA par diagonalisation peut alors être appliqué à chaque sous-problème. L'idée ici est d'observer que la CLA VTI du problème élastique (2.53) basée sur des approximations et des simplifications est égale à la somme des contributions des CLA des sous-problèmes pseudo-acoustiques en onde P et en onde S. Nous reviendrons sur ce point pendant le processus de construction des CLA TTI.

Sous-problème en ondes P

Lorsque $V_s = 0$, les coefficients de la matrice VTI (2.31) se simplifient : $C_{11} = \rho V_p^2(1 + 2\varepsilon)$, $C_{33} = \rho V_p^2$, $C_{44} = 0$, $C_{13} = \rho V_p^2 \sqrt{1 + 2\delta}$. Le système de l'élastodynamique VTI 2D au premier ordre (2.32) conduit alors à un sous-problème pseudo-acoustique VTI en ondes P :

$$\begin{cases} \rho \partial_t v_x &= \partial_x \sigma_{xx} \\ \rho \partial_t v_z &= \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= C_{11} \partial_x v_x + C_{13} \partial_z v_z \\ \partial_t \sigma_{zz} &= C_{13} \partial_x v_x + C_{33} \partial_z v_z \end{cases} \quad (2.54)$$

Soient (ω, ξ) les variables de Fourier associées à (t, z) . Le système (2.54) se réécrit :

$$\begin{cases} -\rho i \omega \widehat{v}_x &= \partial_x \widehat{\sigma}_{xx} \\ -\rho i \omega \widehat{v}_z &= -i \xi \widehat{\sigma}_{zz} \\ -i \omega \widehat{\sigma}_{xx} &= C_{11} \partial_x \widehat{v}_x - C_{13} i \xi \widehat{v}_z \\ -i \omega \widehat{\sigma}_{zz} &= C_{13} \partial_x \widehat{v}_x - C_{33} i \xi \widehat{v}_z \end{cases} \quad (2.55)$$

Après simplifications, le système (2.55) devient :

$$\begin{cases} \partial_x \widehat{\sigma}_{xx} &= -\rho i \omega \widehat{v}_x \\ \widehat{\sigma}_{zz} &= \frac{\rho \omega}{\xi} \widehat{v}_z \\ (C_{11} - \frac{i \xi C_{13}^2}{C_{33} i \xi + \frac{\rho \omega^2}{i \xi}}) \partial_x \widehat{v}_x &= -i \omega \widehat{\sigma}_{xx} \\ \widehat{v}_z &= \frac{C_{13}}{C_{33} i \xi + \frac{\rho \omega^2}{i \xi}} \partial_x \widehat{v}_x \end{cases} \quad (2.56)$$

Le système formé de la première et de la troisième équation de (2.56) conduit à une équation différentielle ordinaire en la variable x de la forme :

$$\partial_x W + MW = 0 \quad (2.57)$$

2.2. Construction d'une CLA VTI par diagonalisation

où $W = (\widehat{\sigma_{xx}}, \widehat{v_x})$ et :

$$M = M_{VTI,2D,P} = \begin{bmatrix} 0 & \rho i \omega \\ i \omega \frac{C_{33}\xi^2 - \rho \omega^2}{(C_{11}C_{33} - C_{13}^2)\xi^2 - C_{11}\rho \omega^2} & 0 \end{bmatrix} \quad (2.58)$$

L'équation (2.57) admet une solution de la forme : $W = W_0 e^{Mx}$ dont les composantes sont telles que $W_i = \sum_{k=1}^4 A_i P_{ik} e^{\Lambda_i x}$, $\forall i = 1, \dots, 4$, avec A_i des constantes à déterminer et (Λ_i, P_i) désignant le couple i -ème couple "valeur propre vecteur propre" de la matrice M définie par (2.58).

Les valeurs propres Λ et la matrice des vecteurs propres associés P sont faciles à déterminer :

$$\Lambda_{VTI,2D,P} = \begin{bmatrix} \lambda_p \\ -\lambda_p \end{bmatrix} \quad P_{VTI,2D,P} = \begin{bmatrix} 1 & -\frac{\rho i \omega}{\lambda_p} \\ \frac{\lambda_p}{\rho i \omega} & 1 \end{bmatrix} \quad (2.59)$$

avec λ_p la valeur propre relative aux ondes P du sous-problème pseudo-acoustique découplé :

$$\lambda_p = \sqrt{\frac{-\rho \omega^2 (C_{33}\xi^2 - \rho \omega^2)}{(C_{11}C_{33} - C_{13}^2)\xi^2 - C_{11}\rho \omega^2}} \quad (2.60)$$

Le domaine de propagation étant défini comme le demi-plan $\{x \leq 0\}$, les solutions portées par des exponentielles à exposant négatif sont à éliminer car elles ne sont pas d'énergie finie. Les solutions de l'équation (2.57) s'écrivent alors :

$$\begin{cases} \widehat{\sigma_{xx}} &= A e^{\lambda_p x} \\ \widehat{v_x} &= A \frac{\lambda_p}{\rho i \omega} e^{\lambda_p x} \end{cases} \quad (2.61)$$

En se plaçant sur la frontière absorbante, c'est-à-dire en $x = 0$, la relation (2.61) se simplifie en :

$$\begin{cases} \widehat{\sigma_{xx}}(0) &= A \\ \widehat{v_x}(0) &= A \frac{\lambda_p}{\rho i \omega} \end{cases} \quad (2.62)$$

Ce qui se conduit à la relation :

$$\rho i \omega \widehat{v_x} = \lambda_p \widehat{\sigma_{xx}} \quad (2.63)$$

Nous localisons la condition en appliquant une approximation de Taylor d'ordre zéro en adoptant la notation " \approx " introduite à la section précédente. Comme le terme $\frac{\xi^2}{\omega^2}$ est très petit, la racine carrée de la valeur propre (2.60) peut alors être approchée suivant la formule (2.17) :

$$\lambda_p = -\frac{\sqrt{\rho} i \omega}{\sqrt{C_{11}}} + O\left(\frac{\xi^2}{\omega^2}\right) \quad (2.64)$$

L'égalité (2.63) se simplifie alors en :

$$\rho i \omega \widehat{v_x} \approx -\frac{\sqrt{\rho} i \omega}{\sqrt{C_{11}}} \widehat{\sigma_{xx}} \quad (2.65)$$

Chapitre 2. Conditions aux limites absorbantes pour des matériaux VTI

Après transformée de Fourier inverse, l'égalité (2.65) donne la CLA ordre faible VTI 2D en ondes P du sous-problème pseudo-acoustique découlé :

$$\sigma_{xx} = -\rho V_p \sqrt{1 + 2\varepsilon v_x} \quad (2.66)$$

Remarque 2.8 La CLA d'ordre faible VTI 2D du sous-problème en ondes P ne fait pas intervenir le coefficient de Thomsen δ .

Sous-problème en ondes S

Lorsque $V_p = 0$, les coefficients de la matrice d'élasticité (2.31) se simplifient. Le système de l'élastodynamique VTI 2D au premier ordre (2.32) conduit alors à un sous-problème pseudo-acoustique en ondes S :

$$\begin{cases} \rho \partial_t v_x &= \partial_x \sigma_{xx} + \partial_z \sigma_{xz} \\ \rho \partial_t v_z &= \partial_x \sigma_{xz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= -2\rho V_s^2 \partial_z v_z \\ \partial_t \sigma_{zz} &= -2\rho V_s^2 \partial_x v_x \\ \partial_t \sigma_{xz} &= \rho V_s^2 (\partial_x v_z + \partial_z v_x) \end{cases} \quad (2.67)$$

Soient (ω, ξ) les variables de Fourier associées à (t, z) . Le système (2.67) se réécrit :

$$\begin{cases} -\rho i \omega \widehat{v}_x &= \partial_x \widehat{\sigma}_{xx} - i \xi \widehat{\sigma}_{xz} \\ -\rho i \omega \widehat{v}_z &= \partial_x \widehat{\sigma}_{xz} - i \xi \widehat{\sigma}_{zz} \\ -i \omega \widehat{\sigma}_{xx} &= 2\rho V_s^2 i \xi \widehat{v}_z \\ -i \omega \widehat{\sigma}_{zz} &= -2\rho V_s^2 \partial_x \widehat{v}_x \\ -i \omega \widehat{\sigma}_{xz} &= \rho V_s^2 (\partial_x \widehat{v}_z - i \xi \widehat{v}_x) \end{cases} \quad (2.68)$$

Après simplifications, les deux premières équations du système (2.68) deviennent :

$$\begin{cases} \widehat{v}_x &= \frac{V_s^2 i \xi}{V_s^2 \xi^2 - \omega^2} \partial_x \widehat{v}_z \\ \widehat{v}_z &= -\frac{V_s^2}{\omega^2} \partial_x^2 \widehat{v}_z - \frac{i \xi V_s^2}{\omega^2} \partial_x \widehat{v}_x \end{cases} \quad (2.69)$$

La substitution de la première équation dans la seconde mène à une équation différentielle ordinaire en la variable x :

$$\partial_x W + M W = 0 \quad (2.70)$$

avec $W = (\partial_x \widehat{v}_z, \widehat{v}_z)$ et :

$$M = M_{VTI,2D,S} = \begin{bmatrix} 0 & \xi^2 - \frac{\omega^2}{V_s^2} \\ -1 & 0 \end{bmatrix} \quad (2.71)$$

L'équation (2.70) admet une solution de la forme : $W = W_0 e^{Mx}$ dont les composantes sont telles que $W_i = \sum_{k=1}^4 A_i P_{ik} e^{\Lambda_i x}$, $\forall i = 1, \dots, 2$, avec A_i des constantes à déterminer et (Λ_i, P_i) désignant le couple i -ème couple "valeur propre vecteur propre" de la matrice M

2.2. Construction d'une CLA VTI par diagonalisation

définie par (2.71).

Les valeurs propres Λ et la matrice des vecteurs propres associés P sont faciles à déterminer :

$$\Lambda_{2D,S} = \begin{bmatrix} \lambda_s \\ -\lambda_s \end{bmatrix} \quad P_{2D,S} = \begin{bmatrix} 1 & -\lambda_s \\ \frac{1}{\lambda_s} & 1 \end{bmatrix} \quad (2.72)$$

avec λ_s la valeur propre relative aux ondes S du sous-problème pseudo-acoustique découplé :

$$\left\{ \begin{array}{l} \lambda_s = \frac{\sqrt{\omega^2 - \xi^2 V_s^2}}{V_s} \end{array} \right. \quad (2.73)$$

Le domaine de propagation étant défini comme le demi-plan $\{x \leq 0\}$, les solutions portées par des exponentielles à exposant négatif sont à éliminer car elles ne sont pas d'énergie finie. Les solutions de l'équation (2.70) s'écrivent alors :

$$\left\{ \begin{array}{l} \partial_x \hat{v}_z = A e^{\lambda_s x} \\ \hat{v}_z = A \frac{1}{\lambda_s} e^{\lambda_s x} \end{array} \right. \quad (2.74)$$

En se plaçant sur la frontière absorbante, c'est-à-dire en $x = 0$, la relation (2.74) se simplifie en :

$$\left\{ \begin{array}{l} \partial_x \hat{v}_z(0) = A \\ \hat{v}_z(0) = A \frac{1}{\lambda_s} \end{array} \right. \quad (2.75)$$

Ce qui se conduit à la relation :

$$\lambda_s \hat{v}_z = \partial_x \hat{v}_z \quad (2.76)$$

Nous localisons la condition en appliquant une approximation de Taylor d'ordre zéro en adoptant la notation “ \approx ” introduite à la section précédente. Comme le terme $\frac{\xi^2}{\omega^2}$ est très petit, la racine carrée de la valeur propre (2.73) peut alors être approchée suivant la formule (2.17) comme pour le sous-problème en onde P et la relation (2.76) devient :

$$\frac{i\omega}{V_s} \hat{v}_z \approx \partial_x \hat{v}_z \quad (2.77)$$

Après transformée de Fourier inverse, l'égalité (2.77) donne une CLA d'ordre faible :

$$\partial_x v_z = -\frac{1}{V_s} \partial_t v_z \quad (2.78)$$

En utilisant l'équation sur σ_{xz} du système (2.67) : $\partial_t \sigma_{xz} = \rho V_s^2 (\partial_x v_z + \partial_z v_x)$, la CLA (2.78) est équivalente à :

$$\partial_t \sigma_{xz} = \rho V_s^2 \partial_z v_x - \rho V_s \partial_t v_z \quad (2.79)$$

Pour terminer, les termes comportant des dérivées en z sont éliminés pour se focaliser sur les ondes de plus fortes amplitudes, conduisant à une CLA VTI en ondes S plus simple pour sous-problème pseudo-acoustique découplé :

$$\sigma_{xz} = -\rho V_s v_z \quad (2.80)$$

Remarque 2.9 *Le sous-problème pseudo-acoustique en onde S 2D est le même pour des configurations isotrope ou VTI. En effet, il ne fait intervenir aucun coefficient de Thomsen.*

2.2.3 Étude expérimentale pour l'élastique VTI 2D

Cas tests

Pour commencer, nous reprenons les cas tests simples déjà décrits dans le cas isotrope, à savoir un cas homogène, c'est-à-dire avec un seul matériau, puis des cas bi-couches, de type bande ou coin, voir Fig. 2.5, de dimension 10 km^2 dont les caractéristiques physiques sont décrites dans le tableau 2.3 (où nous considérons que la masse volumique en 2D est une masse surfacique).

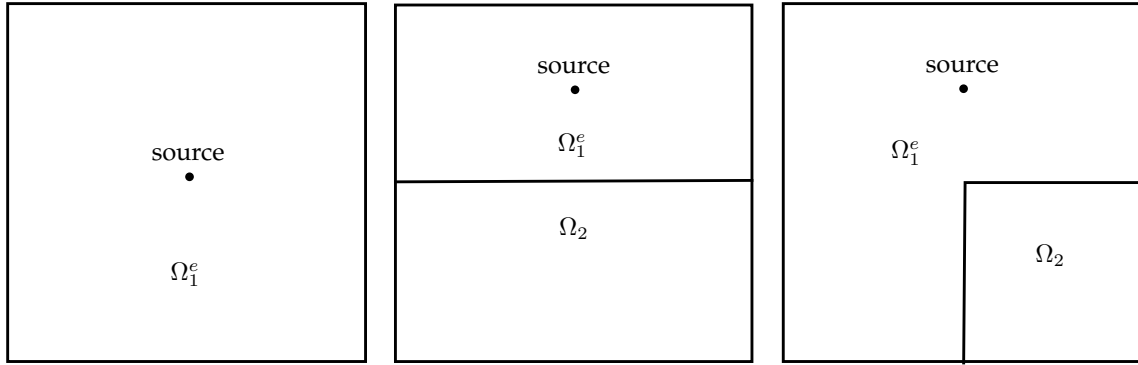


FIGURE 2.5 – Cas tests VTI 2D homogène (gauche), bi-couches (centre) et coin (droite)

milieu	$\rho \text{ [kg.m}^{-2}\text{]}$	$V_p \text{ [m.s}^{-1}\text{]}$	$V_s \text{ [m.s}^{-1}\text{]}$	ε	δ
Ω_1	1	3000	2200	0.20	0.10
Ω_1^e	1	3000	2200	0.20	0.20
Ω_2	1	1800	1100	0.15	0.05

TABLE 2.3 – Caractéristiques physiques des cas-tests VTI

Source et condition initiale

De même que pour le cas isotrope, nous allons suivre la procédure décrite dans la section 1.1.2 pour construire une condition initiale et une source afin de ne générer qu'une onde P, qu'une onde S ou la combinaison d'une onde P et d'une onde S.

En VTI 2D, la matrice de l'équation de dispersion s'écrit :

$$E_{VTI,2D} = \begin{pmatrix} C_{11}k_x^2 + C_{44}k_z^2 & (C_{44} + C_{13})k_xk_z \\ (C_{44} + C_{13})k_xk_z & C_{44}k_x^2 + C_{33}k_z^2 \end{pmatrix} \quad (2.81)$$

où C est la matrice d'élasticité VTI 2D (2.31).

Dans le cas elliptique, les vecteurs propres de (2.81) sont orthogonaux et font intervenir

2.2. Construction d'une CLA VTI par diagonalisation

les symboles de Fourier :

$$\Lambda_p = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} i k_x \\ \sqrt{V_p^2 - V_s^2} i k_z \end{pmatrix}, \quad \Lambda_s = \begin{pmatrix} -\sqrt{V_p^2 - V_s^2} i k_z \\ \sqrt{\kappa^2 V_p^2 - V_s^2} i k_x \end{pmatrix} \quad (2.82)$$

ce qui conduit aux opérateurs :

$$B_p = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_x \\ \sqrt{V_p^2 - V_s^2} \partial_z \end{pmatrix}, \quad B_s = \begin{pmatrix} -\sqrt{V_p^2 - V_s^2} \partial_z \\ \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_x \end{pmatrix} \quad (2.83)$$

Nous rappelons que notre choix de formule pour les conditions initiales est $\underline{\sigma}_0 = 0$ et $\mathbf{v}_0(\mathbf{x}) = B_\star(e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2})$. Pour l'équation élastique VTI elliptique 2D, cela conduit aux différents choix :

$$\left\{ \begin{array}{l} \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} (x - x_0) \\ \sqrt{V_p^2 - V_s^2} (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \text{ pour ne générer qu'une onde P,} \\ \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} -\sqrt{V_p^2 - V_s^2} (z - z_0) \\ \sqrt{\kappa^2 V_p^2 - V_s^2} (x - x_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \text{ pour ne générer qu'une onde S,} \end{array} \right. \quad (2.84)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Pour utiliser une source ponctuelle à la place d'une condition initiale, nous rappelons que sa forme lorsqu'elle est appliquée sur l'équation du tenseur des contraintes est $s_R(t) \underline{s}(\mathbf{x})$, où $s_R(t)$ est une amplitude (en l'occurrence une ondelette de Ricker définie en (1.9)). Il suffit ensuite de chercher \underline{s} telle que $\nabla \cdot \underline{s} = B_\star(\delta_{\mathbf{x}_0})$. Ecrivons cette égalité pour B_p :

$$\begin{pmatrix} \partial_x s_{xx} + \partial_z s_{xz} \\ \partial_x s_{zx} + \partial_z s_{zz} \end{pmatrix} = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_x \\ \sqrt{V_p^2 - V_s^2} \partial_z \end{pmatrix} \delta_{\mathbf{x}_0} \quad (2.85)$$

L'identification terme à terme conduit à une solution simple : $s_{xx} = \sqrt{\kappa^2 V_p^2 - V_s^2} \delta_{\mathbf{x}_0}$, $s_{xz} = s_{zx} = 0$ et $s_{zz} = \sqrt{V_p^2 - V_s^2} \delta_{\mathbf{x}_0}$. En procédant de la même façon pour B_s , les différents choix de sources pour l'équation élastique VTI elliptique 2D s'écrivent finalement :

$$\left\{ \begin{array}{l} \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \\ 0 & \sqrt{V_p^2 - V_s^2} \end{pmatrix}, \text{ pour ne générer qu'une onde P,} \\ \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} 0 & -\sqrt{V_p^2 - V_s^2} \\ \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \end{pmatrix}, \text{ pour ne générer qu'une onde S,} \end{array} \right. \quad (2.86)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Résultats numériques

Comme pour le cas isotrope, sur les trois configurations de domaine décrites Fig. 2.5 et dans le tableau 2.3, nous allons considérer une source élastique (ondes P et S) puis des sources pseudo-acoustiques (onde P ou onde S). Les résultats sur le domaine homogène sont présentés Fig. 2.6, ceux des domaines bi-couches bande et coin sont présentés Fig. 2.7 et 2.8 respectivement. Le module du vecteur vitesse est représenté pour différents temps de simulation. À $t = 1s$, nous observons le front d'onde, ce qui permet de figer l'échelle des représentations. À $t = 2s$ ou $t = 3s$, nous commençons à observer le travail des CLA car le front d'onde a atteint les bords du domaine de calcul. Nous laissons ensuite les calculs continuer jusqu'à $t = 5s$ puis $t = 7s$. Nous observons ainsi les réflexions parasites propres aux bords absorbants. Les mêmes résultats sont ensuite présentés à l'échelle de couleurs des réflexions, calculée au temps, en fonction des cas, $t = 5s$ (homogène) ou $t = 7s$ (bi-couches). Toutes les valeurs des échelles sont regroupées dans le tableau 2.4.

De même qu'en isotrope, nous observons un comportement connu : plus l'angle d'incidence de l'onde s'éloigne de l'axe normal, plus les réflexions sont fortes. C'est tout à fait cohérent avec le mode d'approximation qui a été choisi pour localiser la condition aux limites transparente, qui est le même que pour le cas isotrope. De plus, les ondes S génèrent des réflexions plus fortes que celles des ondes P, ce qui justifie aussi le choix de considérer une source en onde P seule, pour mieux étudier les réflexions et se rapprocher des modèles pseudo-acoustiques utilisés par les géophysiciens.

source	temps	homogène	bi-couche bande	bi-couche coin
PS	1s / 1s / 1s	2.00	2.00	2.00
	5s / 7s / 7s	0.12	0.29	0.33
P	1s / 1s / 1s	1.68	1.68	1.68
	5s / 7s / 7s	0.04	0.12	0.06
S	1s / 1s / 1s	2.00	2.00	2.00
	5s / 7s / 7s	0.12	0.30	0.29

TABLE 2.4 – Valeurs maximales du module du vecteur vitesse pour l'étude VTI 2D

Remarque 2.10 *Comme dit précédemment, ces CLA VTI 2D ont déjà été explicitées, par exemple dans [Bec91]. Cependant, à notre connaissance, il n'y a pas de référence ni sur des résultats numériques, ni sur la construction d'une source ou d'une condition initiale VTI (elliptique ou non). Comme les PML sont stables dans beaucoup de configurations VTI, elles sont privilégiées car plus précises. Nous pouvons néanmoins comparer nos résultats VTI avec nos résultats isotropes.*

Comparaison avec la CLA isotrope

Nous proposons maintenant de comparer la CLA VTI à la CLA isotrope dans le domaine homogène VTI elliptique. Les résultats avec des sources élastiques (ondes P et S) et pseudo-

2.2. Construction d'une CLA VTI par diagonalisation

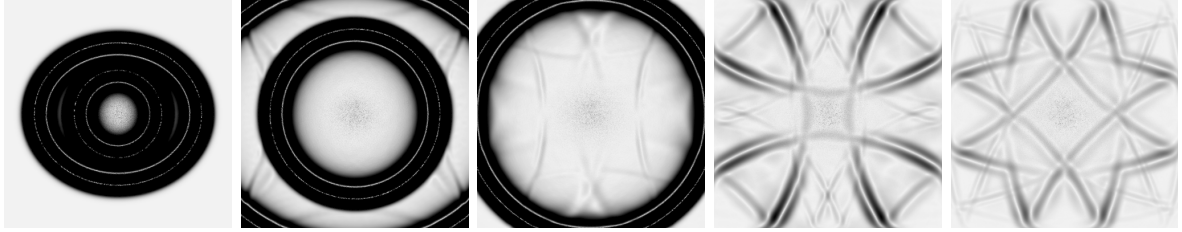
acoustiques (onde P et onde S) sont représentés pour la CLA isotrope Fig. 2.9 (à comparer avec la Fig. 2.6). Il apparaît avant toute chose que les réflexions sont assez faibles même en utilisant une CLA isotrope. La plus grosse différence réside dans la réflexion de l'onde P en onde P qui n'apparaît pas dans le cas de la CLA VTI et qui est très présente dans le cas de la CLA isotrope. La réflexion d'onde P en onde S est aussi plus forte avec une CLA isotrope, ceci peut se vérifier en comparant les échelles de représentation des calculs. Enfin, les réflexions de l'onde S sont comparables et cela est logique puisque la partie concernant l'onde S dans la CLA est la même.

Non-elliptique

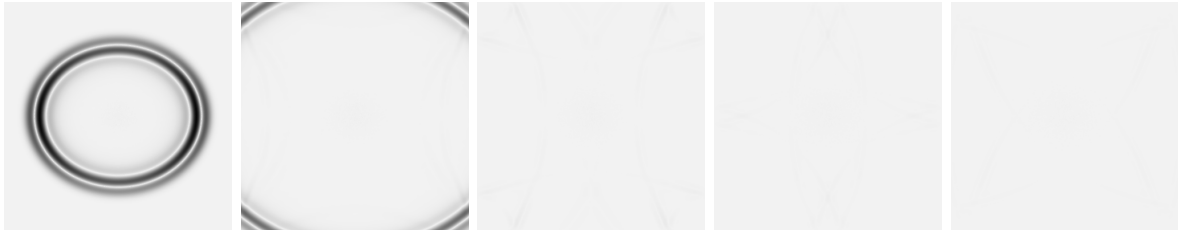
Un exemple de configuration non-elliptique est proposé dans un domaine homogène avec les paramètres précisés dans le tableau 2.3. Les résultats sont présentés pour une source élastique Fig. 2.10. La comparaison avec l'utilisation d'une CLA isotrope est présentée Fig. 2.11. De même que pour le cas elliptique, les réflexions de l'onde P sont beaucoup plus fortes avec une CLA isotrope.



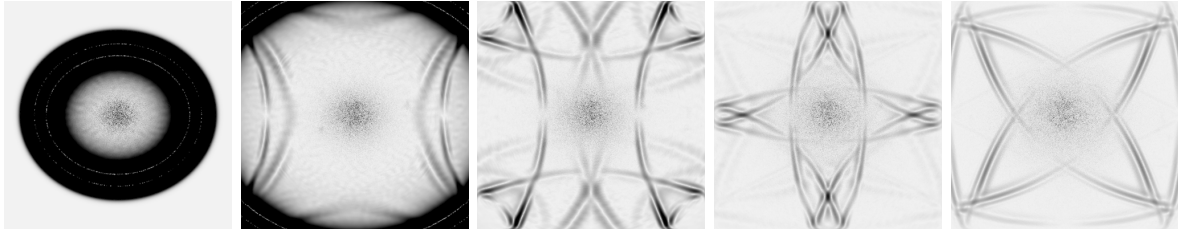
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$



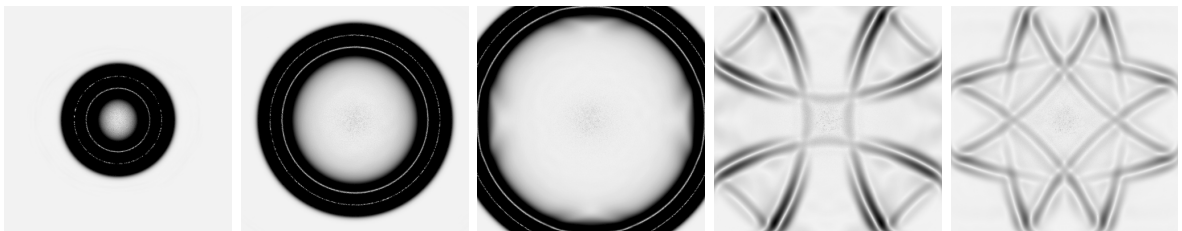
(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 5s$



(e) source S, échelle de couleur fixée à $t = 1s$

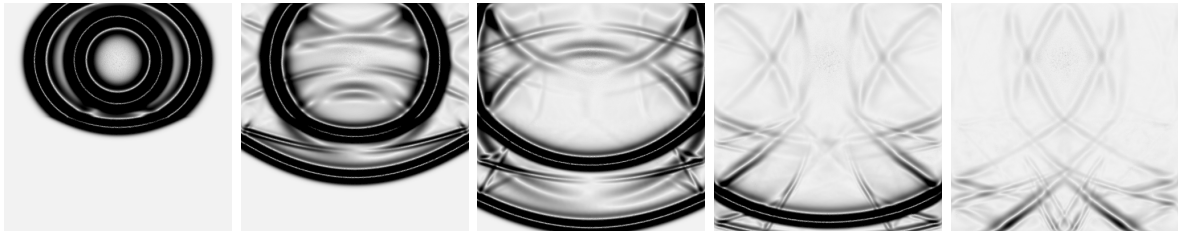


(f) source S, échelle de couleur fixée à $t = 5s$

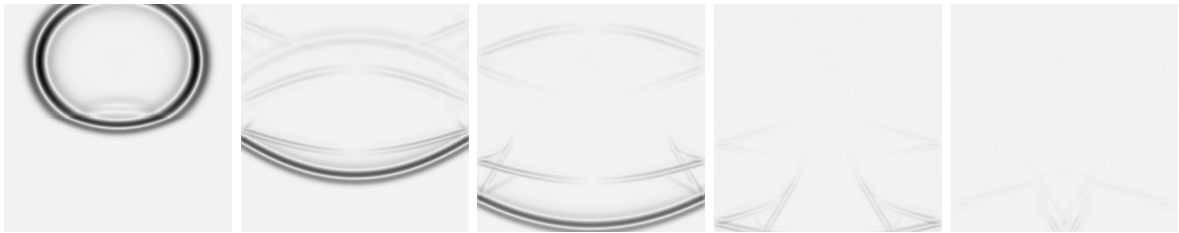
2.2. Construction d'une CLA VTI par diagonalisation



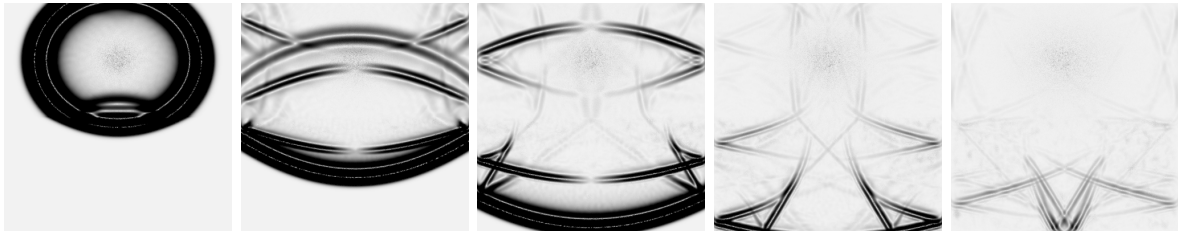
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 7s$



(e) source S, échelle de couleur fixée à $t = 1s$

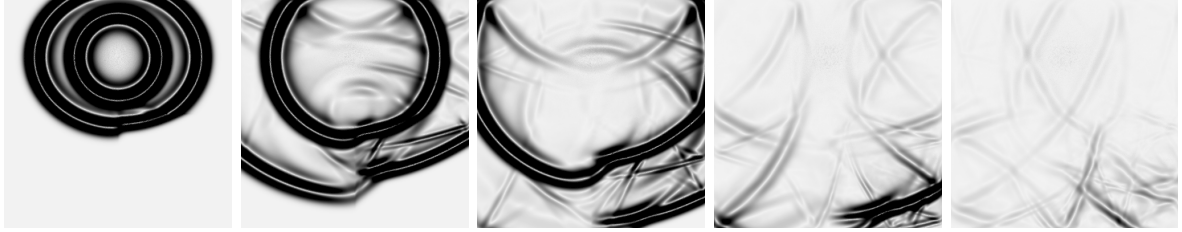


(f) source S, échelle de couleur fixée à $t = 7s$

FIGURE 2.7 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration bi-couche bande VTI 2D



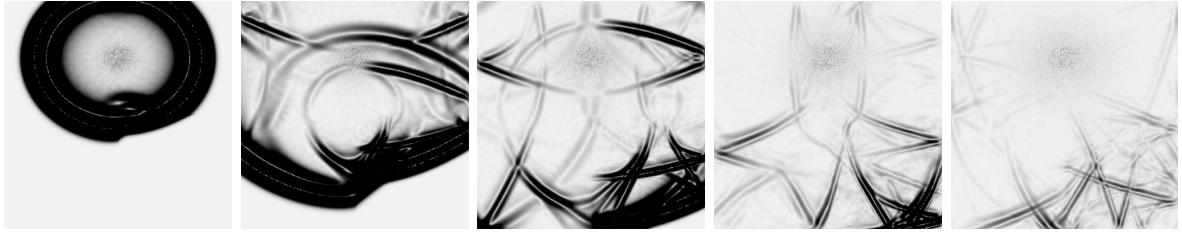
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 7s$



(e) source S, échelle de couleur fixée à $t = 1s$

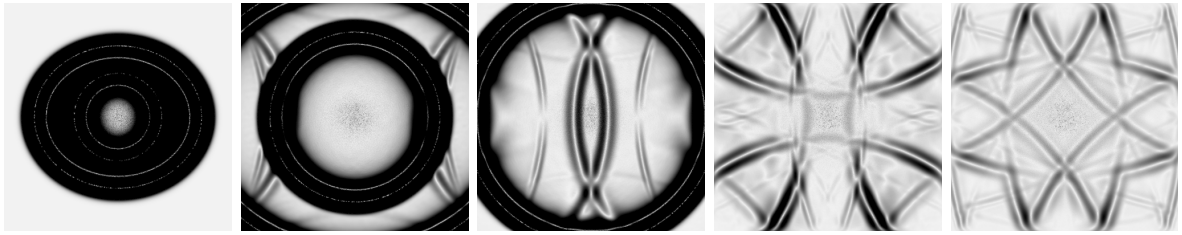


(f) source S, échelle de couleur fixée à $t = 7s$

2.2. Construction d'une CLA VTI par diagonalisation



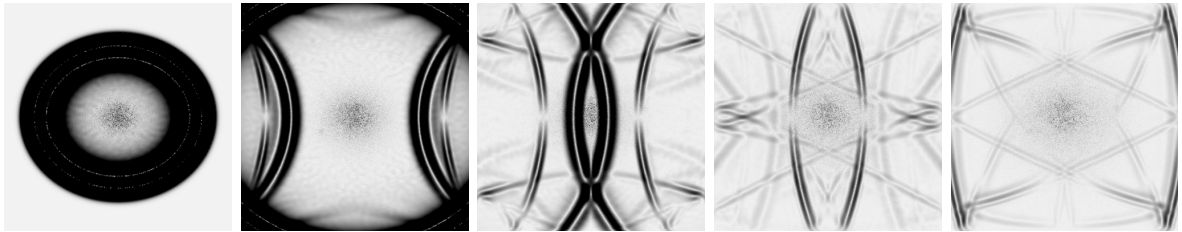
(a) source PS, échelle de couleur fixée à $t = 1s : 2.00$



(b) source PS, échelle de couleur fixée à $t = 5s : 0.15$



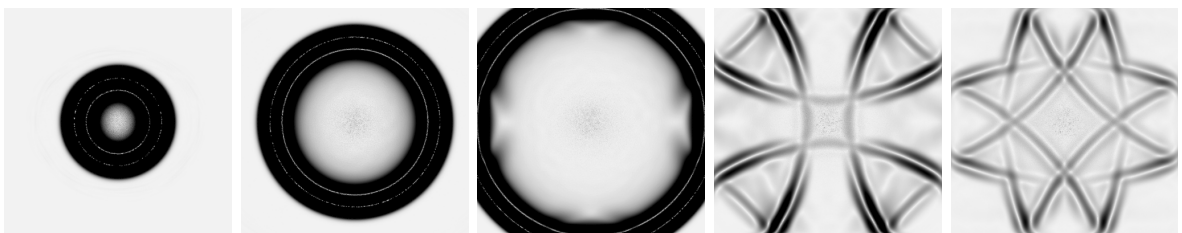
(c) source P, échelle de couleur fixée à $t = 1s : 1.68$



(d) source P, échelle de couleur fixée à $t = 5s : 0.04$



(e) source S, échelle de couleur fixée à $t = 1s : 2.00$

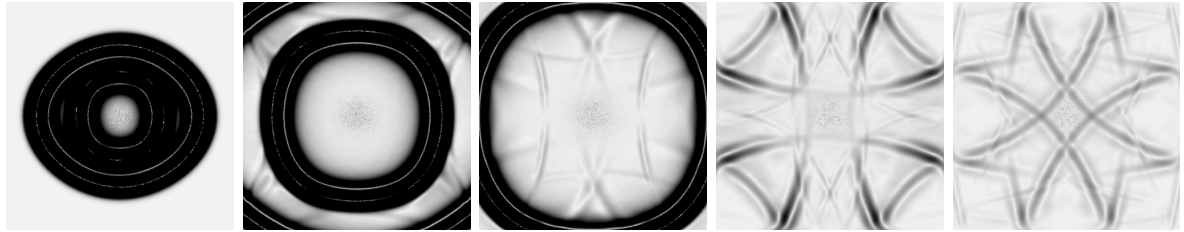


(f) source S, échelle de couleur fixée à $t = 5s : 0.14$

FIGURE 2.9 – Comparaison avec la CLA isotrope à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration homogène VTI 2D



(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$

FIGURE 2.10 – Module du vecteur vitesse au cours du temps à $t = 1s$, $2s$, $5s$ et $7s$ pour un cas 2D VTI non-elliptique homogène. Les valeurs maximales sont 2.04 et 0.11 aux temps respectifs $t = 1s$ et $t = 5s$.



(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$

FIGURE 2.11 – Comparaison avec la CLA isotrope à $t = 1s$, $2s$, $5s$ et $7s$ pour un cas 2D VTI non-elliptique homogène. Les valeurs maximales sont 2.04 et 0.13 aux temps respectifs $t = 1s$ et $t = 5s$.

2.3 Stabilité de la CLA VTI 2D

2.3.1 Stabilité des problèmes mixtes continu, semi-discret et discret

Énergie associée au problème mixte continu

Nous rappelons que les paramètres physiques sont définis par $\rho > 0$, $V_p > 0$, $V_s > 0$ et $\underline{\underline{C}}$ symétrique et positif, ie $\forall \underline{\underline{\xi}}$ symétrique, $(\underline{\underline{C}} \underline{\underline{\xi}}) : \underline{\underline{\xi}} > 0$. Le tenseur $\underline{\underline{C}}$ est donc inversible et nous noterons $\underline{\underline{C}}^{-1}$ son inverse, qui est aussi positif.

Pour étudier la stabilité de la CLA dans le cas continu, introduisons le domaine $\Omega_x = \{x : x \in]-\infty, 0]\}$ et Γ_x sa frontière définie par $\{x = 0\}$. Soit la fonction $\mathbb{E}(t) = \frac{1}{2} \left(\|\mathbf{v}\|_{\Omega_x, \rho}^2 + \|\underline{\underline{\sigma}}\|_{\Omega_x, \underline{\underline{C}}^{-1}}^2 \right)$. Alors, formellement :

$$\frac{d}{dt} \mathbb{E}(t) = \int_{\Omega_x} \rho \partial_t \mathbf{v} \cdot \mathbf{v} d\mathbf{x} + \int_{\Omega_x} \partial_t \underline{\underline{\sigma}} : (\underline{\underline{C}}^{-1} \underline{\underline{\sigma}}) d\mathbf{x} \quad (2.87)$$

Il est possible de modifier la première intégrale de (2.87) en utilisant la première équation du système de l'élastodynamique (1.5) puis en appliquant une intégration par partie (formule de Green). Il vient :

$$\int_{\Omega_x} \rho \partial_t \mathbf{v} \cdot \mathbf{v} d\mathbf{x} = \int_{\Omega_x} (\nabla \cdot \underline{\underline{\sigma}}) \cdot \mathbf{v} d\mathbf{x} = - \int_{\Omega_x} \underline{\underline{\sigma}} : \nabla \mathbf{v} d\mathbf{x} + \int_{\Gamma_x} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{v} d\mathbf{x} \quad (2.88)$$

De même, la deuxième intégrale de (2.87) peut être modifiée en utilisant la deuxième équation du système de l'élastodynamique (1.5) et le fait que $\forall \underline{\underline{\xi}}$, $(\underline{\underline{C}} \underline{\underline{\epsilon}}(\mathbf{v})) : \underline{\underline{\xi}} = (\underline{\underline{C}} \underline{\underline{\xi}}) : \nabla \mathbf{v}$. Nous obtenons :

$$\int_{\Omega_x} \partial_t \underline{\underline{\sigma}} : (\underline{\underline{C}}^{-1} \underline{\underline{\sigma}}) d\mathbf{x} = \int_{\Omega_x} (\underline{\underline{C}} \underline{\underline{\epsilon}}(\mathbf{v})) : (\underline{\underline{C}}^{-1} \underline{\underline{\sigma}}) d\mathbf{x} = \int_{\Omega_x} \underline{\underline{\sigma}} : \nabla \mathbf{v} d\mathbf{x} \quad (2.89)$$

En injectant les deux intégrales (2.88) et (2.89) dans l'équation (2.87), il vient alors :

$$\frac{d}{dt} \mathbb{E}(t) = \int_{\Gamma_x} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{v} d\mathbf{x} \quad (2.90)$$

Le vecteur \mathbf{n} désigne la normale extérieure à Γ_x donc $\mathbf{n} = (1, 0)$ et $\underline{\underline{\sigma}} \mathbf{n} = (\sigma_{xx}, \sigma_{xz})$. Enfin, sur la frontière Γ_x , les composantes de $\underline{\underline{\sigma}}$ sont définies par la CLA VTI du système (2.53). Ainsi :

$$\frac{d}{dt} \mathbb{E}(t) = - \int_{\Gamma_x} (\rho V_p \sqrt{1 + 2\varepsilon v_x^2} + \rho V_s v_z^2) d\mathbf{x} \quad (2.91)$$

Comme cette quantité est négative, l'énergie du système continu est décroissante, ce qui assure la stabilité du problème de l'élastodynamique 2D VTI avec la CLA (2.53).

Énergie associée au problème mixte semi-discret

En reprenant les développements de la sous-section 1.3.2 sur la discrétisation par DGM, la formulation variationnelle (1.36) du système de l'élastodynamique au premier ordre (1.5),

peut s'écrire sous la forme :

$$\left\{ \begin{aligned} \sum_{K \in \mathcal{T}_h} \int_K \rho \partial_t \underline{\mathbf{v}} \cdot \underline{\mathbf{w}} d\mathbf{x} &= - \sum_{K \in \mathcal{T}_h} \int_K \underline{\underline{\underline{\sigma}}} : \nabla \underline{\mathbf{w}} d\mathbf{x} \\ &+ \sum_{K \in \mathcal{T}_h} \left(\sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \{ \{ \underline{\underline{\underline{\sigma}}} \} \} \llbracket \underline{\mathbf{w}} \rrbracket \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} \underline{\underline{\underline{\sigma}}} \underline{\mathbf{n}} \cdot \underline{\mathbf{w}} d\mathbf{x} \right) \\ \sum_{K \in \mathcal{T}_h} \int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} &= - \sum_{K \in \mathcal{T}_h} \int_K \left(\nabla \cdot (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \right) \cdot \underline{\mathbf{v}} d\mathbf{x} \\ &+ \sum_{K \in \mathcal{T}_h} \left(\sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \llbracket (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \rrbracket \{ \{ \underline{\mathbf{v}} \} \} \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \cdot \underline{\mathbf{n}} d\mathbf{x} \right) \end{aligned} \right. \quad (2.92)$$

Ce schéma est utilisé dans le code DIVA de Total et, numériquement, les calculs sont faits élément par élément. Pour les termes de flux, les calculs sont partagés par rapport à la fonction-test $\underline{\mathbf{w}}$ ou $\underline{\underline{\underline{\xi}}}$. La seconde équation de (2.92) s'écrit alors pour un élément K :

$$\int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} = - \int_K \left(\nabla \cdot (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \right) \cdot \underline{\mathbf{v}} d\mathbf{x} + \int_{\partial K \setminus \Gamma_{out}} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \{ \{ \underline{\mathbf{v}} \} \} \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \cdot \underline{\mathbf{n}} d\mathbf{x}$$

En développant, il vient :

$$\int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} = - \int_K \left(\nabla \cdot (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \right) \cdot \underline{\mathbf{v}} d\mathbf{x} + \int_{\partial K \setminus \Gamma_{out}} \frac{1}{2} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) (\underline{\mathbf{v}} + \underline{\mathbf{v}}^{ext}) \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \cdot \underline{\mathbf{n}} d\mathbf{x}$$

où le terme $\underline{\mathbf{v}}^{ext}$ correspond à $\underline{\mathbf{v}}$ sur l'élément voisin à K par rapport à chaque face de ∂K . Puis, en faisant l'intégration par partie inverse, nous avons :

$$\int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} = \int_{\partial K \setminus \Gamma_{out}} \frac{1}{2} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) (\underline{\mathbf{v}} + \underline{\mathbf{v}}^{ext}) \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_K (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) : \nabla \underline{\mathbf{v}} d\mathbf{x} - \int_{\partial K} (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \cdot \underline{\mathbf{n}} d\mathbf{x}$$

Les termes en $\underline{\mathbf{v}}$ se regroupent alors :

$$\int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} = - \int_{\partial K \setminus \Gamma_{out}} \frac{1}{2} \llbracket (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \rrbracket \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_K (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) : \nabla \underline{\mathbf{v}} d\mathbf{x}$$

Nous rappelons que par continuité : $\llbracket (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \underline{\mathbf{v}} \rrbracket \cdot \underline{\mathbf{n}} = \llbracket (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \rrbracket \{ \{ \underline{\mathbf{v}} \} \} \cdot \underline{\mathbf{n}}$, et donc finalement, en sommant :

$$\sum_{K \in \mathcal{T}_h} \int_K \partial_t \underline{\underline{\underline{\sigma}}} : \underline{\underline{\underline{\xi}}} d\mathbf{x} = \sum_{K \in \mathcal{T}_h} \left(- \sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \llbracket (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) \rrbracket \{ \{ \underline{\mathbf{v}} \} \} \cdot \underline{\mathbf{n}} d\mathbf{x} + \int_K (\underline{\underline{\underline{C}}} \underline{\underline{\underline{\xi}}}) : \nabla \underline{\mathbf{v}} d\mathbf{x} \right)$$

Le schéma numérique (2.92) est ainsi équivalent à :

$$\left\{ \begin{array}{l} \sum_{K \in \mathcal{T}_h} \int_K \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} = - \sum_{K \in \mathcal{T}_h} \int_K \underline{\underline{\sigma}} : \nabla \mathbf{w} d\mathbf{x} \\ \quad + \sum_{K \in \mathcal{T}_h} \left(\sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \{ \{ \underline{\underline{\sigma}} \} \} \llbracket \mathbf{w} \rrbracket \cdot \mathbf{n} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} \underline{\underline{\sigma}} \mathbf{n} \cdot \mathbf{w} d\mathbf{x} \right) \\ \sum_{K \in \mathcal{T}_h} \int_K \partial_t \underline{\underline{\sigma}} : \underline{\underline{\xi}} d\mathbf{x} = \sum_{K \in \mathcal{T}_h} \int_K (\underline{\underline{C}} \underline{\underline{\xi}}) : \nabla \mathbf{v} d\mathbf{x} \\ \quad - \sum_{K \in \mathcal{T}_h} \sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \llbracket (\underline{\underline{C}} \underline{\underline{\xi}}) \rrbracket \{ \{ \mathbf{v} \} \} \cdot \mathbf{n} d\mathbf{x} \end{array} \right. \quad (2.93)$$

Le tenseur $\underline{\underline{C}}$ étant inversible, nous introduisons $\tilde{\underline{\underline{\xi}}} = \underline{\underline{C}}^{-1} \underline{\underline{\xi}}$ et soit C^{-1} la notation matricielle de $\underline{\underline{C}}^{-1}$. Le système (2.93) est vrai quel que soit le couple $(\mathbf{w}, \underline{\underline{\xi}})$, et donc notamment pour $(\mathbf{w}, \tilde{\underline{\underline{\xi}}})$:

$$\left\{ \begin{array}{l} \sum_{K \in \mathcal{T}_h} \int_K \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} = - \sum_{K \in \mathcal{T}_h} \int_K \underline{\underline{\sigma}} : \nabla \mathbf{w} d\mathbf{x} \\ \quad + \sum_{K \in \mathcal{T}_h} \left(\sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \{ \{ \underline{\underline{\sigma}} \} \} \llbracket \mathbf{w} \rrbracket \cdot \mathbf{n} d\mathbf{x} + \int_{\partial K \cap \Gamma_{out}} \underline{\underline{\sigma}} \mathbf{n} \cdot \mathbf{w} d\mathbf{x} \right) \\ \sum_{K \in \mathcal{T}_h} \int_K \partial_t \underline{\underline{\sigma}} : (\underline{\underline{C}}^{-1} \underline{\underline{\xi}}) d\mathbf{x} = \sum_{K \in \mathcal{T}_h} \int_K \underline{\underline{\xi}} : \nabla \mathbf{v} d\mathbf{x} \\ \quad - \sum_{K \in \mathcal{T}_h} \sum_{L \in \mathcal{T}_h} \int_{\partial K \cap \partial L \setminus \Gamma_{out}} \llbracket \underline{\underline{\xi}} \rrbracket \{ \{ \mathbf{v} \} \} \cdot \mathbf{n} d\mathbf{x} \end{array} \right. \quad (2.94)$$

ce qui se réécrit :

$$\left\{ \begin{array}{l} \int_{\Omega_h} \rho \partial_t \mathbf{v} \cdot \mathbf{w} d\mathbf{x} = \int_{\Gamma_{in}} \{ \{ \underline{\underline{\sigma}} \} \} \llbracket \mathbf{w} \rrbracket \cdot \mathbf{n} d\mathbf{x} + \int_{\Gamma_{out}} (\underline{\underline{\sigma}} \mathbf{n}) \cdot \mathbf{w} d\mathbf{x} - \int_{\Omega_h} \underline{\underline{\sigma}} : \nabla_h \mathbf{w} d\mathbf{x} \\ \int_{\Omega_h} \partial_t \underline{\underline{\sigma}} : (\underline{\underline{C}}^{-1} \underline{\underline{\xi}}) d\mathbf{x} = - \int_{\Gamma_{in}} \llbracket \underline{\underline{\xi}} \rrbracket \{ \{ \mathbf{v} \} \} \cdot \mathbf{n} d\mathbf{x} + \int_{\Omega_h} \underline{\underline{\xi}} : \nabla_h \mathbf{v} d\mathbf{x} \end{array} \right. \quad (2.95)$$

En reprenant les notations matricielles de la sous-section 1.3.2, le système (2.95) s'écrit :

$$\left\{ \begin{array}{l} (a) \quad M_{\mathbf{v}} \partial_t \mathbf{v}_h + R_{\sigma} \sigma_h + B_{\mathbf{v}} \mathbf{v}_h = 0 \\ (b) \quad M_{\sigma, C^{-1}} \partial_t \sigma_h - R_{\mathbf{v}, C^{-1}} \mathbf{v}_h = 0 \end{array} \right. \quad (2.96)$$

où les matrices $M_{\mathbf{v}}$, R_{σ} et $B_{\mathbf{v}}$ sont définies dans (1.43), (1.44) et (1.45). Les matrices $M_{\sigma, C^{-1}}$, et $R_{\mathbf{v}, C^{-1}}$ se construisent comme les matrices M_{σ} et $R_{\mathbf{v}}$ définies dans (1.46) et (1.47) mais les paramètres physiques ne sont plus factorisés dans la matrice de rigidité mais dans la matrice de masse. De plus, les termes de bord n'apparaissent plus dans $R_{\mathbf{v}, C^{-1}}$. Ainsi, nous

Chapitre 2. Conditions aux limites absorbantes pour des matériaux VTI

avons simplement $\boxed{R_{\mathbf{v},C^{-1}} = (R_{\boldsymbol{\sigma}})^T}$. La matrice $M_{\boldsymbol{\sigma},C^{-1}}$ est telle que chacune de ses matrices $M_{\boldsymbol{\sigma},C^{-1}}^K$ a ses blocs multipliés par le coefficient correspondant de la matrice C^{-1} :

$$(M_{\boldsymbol{\sigma},C^{-1}})_{ij}^K = (C^{-1})_{ij} M_K, \quad \forall i, j = 1..d(d+1)/2$$

où M_K est la matrice de masse sur les fonction-test définie en (1.38). De plus, la matrice $M_{\boldsymbol{\sigma},C^{-1}}$ est inversible et symétrique car les matrices C^{-1} et M_K sont définies positives et symétriques.

En faisant l'opération $\{<(a), \mathbf{v}_h> + <(b), \boldsymbol{\sigma}_h>\}$ dans le système (2.96), il vient :

$$<\partial_t M_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h> + <\partial_t M_{\boldsymbol{\sigma},C^{-1}} \boldsymbol{\sigma}_h, \boldsymbol{\sigma}_h> = - <B_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h> \quad (2.97)$$

De la même manière que pour l'étude de la stabilité du problème continu, il est possible de montrer que $B_{\mathbf{v}}$ est positive en s'appuyant sur les paramètres physiques et en utilisant la structure de la matrice. Le terme de droite dans (2.97) est donc négatif. Par ailleurs, les matrices de masses sont positives. Ceci implique donc une décroissance de l'énergie semi-discrète $\mathbb{E}_{semi_d}(t)$ en posant par exemple :

$$\mathbb{E}_{semi_d}(t) = <\partial_t M_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h> + <\partial_t M_{\boldsymbol{\sigma},C^{-1}} \boldsymbol{\sigma}_h, \boldsymbol{\sigma}_h> \quad (2.98)$$

Énergie associée au problème mixte discret

Pour compléter la discrétisation, le schéma Leap-Frog est appliqué au système (2.96) :

$$\begin{cases} (c) & M_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^n}{\Delta t} + R_{\boldsymbol{\sigma}} \boldsymbol{\sigma}_h^{n+\frac{1}{2}} + B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} = 0 \\ (d) & M_{\boldsymbol{\sigma},C^{-1}} \frac{\boldsymbol{\sigma}_h^{n+\frac{3}{2}} - \boldsymbol{\sigma}_h^{n+\frac{1}{2}}}{\Delta t} - (R_{\boldsymbol{\sigma}})^T \mathbf{v}_h^{n+1} = 0 \end{cases} \quad (2.99)$$

La deuxième équation de (2.99) peut être écrite à un pas de temps précédent :

$$(e) \quad M_{\boldsymbol{\sigma},C^{-1}} \frac{\boldsymbol{\sigma}_h^{n+\frac{1}{2}} - \boldsymbol{\sigma}_h^{n-\frac{1}{2}}}{\Delta t} - (R_{\boldsymbol{\sigma}})^T \mathbf{v}_h^{n+1} = 0 \quad (2.100)$$

À l'aide de (2.99) et (2.100), l'opération $\left\{ <(c), \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2}> + <\frac{(d)+(e)}{2}, \boldsymbol{\sigma}_h^{n+\frac{1}{2}}> \right\}$ donne :

$$\begin{cases} < M_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^n}{\Delta t}, \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} > + < B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2}, \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} > \\ + \frac{1}{2} \left(< M_{\boldsymbol{\sigma},C^{-1}} \frac{\boldsymbol{\sigma}_h^{n+\frac{3}{2}} - \boldsymbol{\sigma}_h^{n+\frac{1}{2}}}{\Delta t}, \boldsymbol{\sigma}_h^{n+\frac{1}{2}} > + < M_{\boldsymbol{\sigma},C^{-1}} \frac{\boldsymbol{\sigma}_h^{n+\frac{1}{2}} - \boldsymbol{\sigma}_h^{n-\frac{1}{2}}}{\Delta t}, \boldsymbol{\sigma}_h^{n+\frac{1}{2}} > \right) = 0, \end{cases} \quad (2.101)$$

ce qui est équivalent à :

$$\left\{ \begin{array}{l} \frac{1}{2\Delta t} < M_{\mathbf{v}} \mathbf{v}_h^{n+1}, \mathbf{v}_h^{n+1} > - \frac{1}{2\Delta t} < M_{\mathbf{v}} \mathbf{v}_h^n, \mathbf{v}_h^n > \\ + \frac{1}{2\Delta t} < M_{\sigma, C^{-1}} \sigma_h^{n+\frac{3}{2}}, \sigma_h^{n+\frac{1}{2}} > - \frac{1}{2\Delta t} < M_{\sigma, C^{-1}} \sigma_h^{n-\frac{1}{2}}, \sigma_h^{n+\frac{1}{2}} > \\ = - < B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2}, \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} > \end{array} \right. \quad (2.102)$$

Nous obtenons donc

$$\frac{\mathbb{E}_d^{n+1} - \mathbb{E}_d^n}{\Delta t} = - < B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2}, \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} > \quad (2.103)$$

où \mathbb{E}_d^n est définie par :

$$\mathbb{E}_d^n = \frac{1}{2} < M_{\mathbf{v}} \mathbf{v}_h^n, \mathbf{v}_h^n > + \frac{1}{2} < M_{\sigma, C^{-1}} \sigma_h^{n+\frac{1}{2}}, \sigma_h^{n-\frac{1}{2}} > \quad (2.104)$$

Nous avons utilisé le fait que la matrice $M_{\sigma, C^{-1}}$ est symétrique, ce qui implique l'égalité :

$$< M_{\sigma, C^{-1}} \sigma_h^{n-\frac{1}{2}}, \sigma_h^{n+\frac{1}{2}} > = < M_{\sigma, C^{-1}} \sigma_h^{n+\frac{1}{2}}, \sigma_h^{n-\frac{1}{2}} > \quad (2.105)$$

Comme $B_{\mathbf{v}}$ est positive, (2.103) est négatif. Nous proposons de définir \mathbb{E}_d^n comme l'énergie discrète du problème mais avant cela, il nous faut vérifier que \mathbb{E}_d^n est positive pour tout n .

En développant par exemple le terme sur la variable du tenseur des contraintes :

$$\left\{ \begin{array}{l} < M_{\sigma, C^{-1}} \sigma_h^{n+\frac{1}{2}}, \sigma_h^{n-\frac{1}{2}} > = < M_{\sigma, C^{-1}} \frac{\sigma_h^{n+\frac{1}{2}} + \sigma_h^{n-\frac{1}{2}}}{2}, \frac{\sigma_h^{n+\frac{1}{2}} + \sigma_h^{n-\frac{1}{2}}}{2} > \\ - \frac{\Delta t^2}{4} < M_{\sigma, C^{-1}} \frac{\sigma_h^{n+\frac{1}{2}} - \sigma_h^{n-\frac{1}{2}}}{\Delta t}, \frac{\sigma_h^{n+\frac{1}{2}} - \sigma_h^{n-\frac{1}{2}}}{\Delta t} > \end{array} \right. \quad (2.106)$$

puis en se servant de l'équation (2.100), il vient par substitution :

$$< M_{\sigma, C^{-1}} \frac{\sigma_h^{n+\frac{1}{2}} - \sigma_h^{n-\frac{1}{2}}}{\Delta t}, \frac{\sigma_h^{n+\frac{1}{2}} - \sigma_h^{n-\frac{1}{2}}}{\Delta t} > = < (R_{\sigma})^T \mathbf{v}_h^n, (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T \mathbf{v}_h^n > \quad (2.107)$$

Nous obtenons donc

$$\left\{ \begin{array}{l} \mathbb{E}_d^n = \frac{1}{2} < \left(M_{\mathbf{v}} - \frac{\Delta t^2}{4} R_{\sigma} (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T \right) \mathbf{v}_h^n, \mathbf{v}_h^n > \\ + \frac{1}{2} < M_{\sigma, C^{-1}} \frac{\sigma_h^{n+\frac{1}{2}} + \sigma_h^{n-\frac{1}{2}}}{2}, \frac{\sigma_h^{n+\frac{1}{2}} + \sigma_h^{n-\frac{1}{2}}}{2} > \end{array} \right. \quad (2.108)$$

\mathbb{E}_d^n est donc positive et décroissante si et seulement si la matrice

$$M_{\mathbf{v}} - \frac{\Delta t^2}{4} R_{\sigma} (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T \quad (2.109)$$

est positive.

Ceci signifie que pour tout vecteur non nul \mathbf{u} nous ayons :

$$< \left(M_{\mathbf{v}} - \frac{\Delta t^2}{4} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T \right) \mathbf{u}, \mathbf{u} > \geq 0$$

ce qui est équivalent à

$$< \left(M_{\mathbf{v}}^{\frac{1}{2}} - \frac{\Delta t^2}{4} M_{\mathbf{v}}^{-\frac{1}{2}} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T M_{\mathbf{v}}^{-\frac{1}{2}} M_{\mathbf{v}}^{\frac{1}{2}} \right) \mathbf{u}, M_{\mathbf{v}}^{\frac{1}{2}} \mathbf{u} > \geq 0$$

car $M_{\mathbf{v}}$ est symétrique définie positive. Puis, en posant $\tilde{\mathbf{u}} = M_{\mathbf{v}}^{\frac{1}{2}} \mathbf{u}$, il vient

$$< \left(I - \frac{\Delta t^2}{4} M_{\mathbf{v}}^{-\frac{1}{2}} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T M_{\mathbf{v}}^{-\frac{1}{2}} \right) \tilde{\mathbf{u}}, \tilde{\mathbf{u}} > \geq 0$$

où I représente la matrice identité.

Comme la matrice $M_{\mathbf{v}}^{-\frac{1}{2}} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T M_{\mathbf{v}}^{-\frac{1}{2}}$ est symétrique, elle est diagonalisable. De plus, ses valeurs propres sont positives ou nulles de part les propriétés des matrices qui la composent. Soit λ_{max} la plus grande de ses valeurs propres, l'inégalité qui doit être vérifiée pour que \mathbb{E}_d^n soit une énergie discrète est

$$1 - \frac{\Delta t^2}{4} \lambda_{max} \geq 0 \quad (2.110)$$

ce qui représente une condition CFL reliant le pas de temps Δt à la discrétisation en espace cachée dans la valeur propre.

En observant maintenant que toute valeur propre de $M_{\mathbf{v}}^{-\frac{1}{2}} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T M_{\mathbf{v}}^{-\frac{1}{2}}$ est aussi valeur propre de $M_{\mathbf{v}}^{-1} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T$, cette condition CFL de stabilité du schéma numérique peut être formellement calculée pour un maillage cartésien mais pas pour un maillage non-structuré dans lequel trop de paramètres entre en jeu. Dans la pratique, la plus grande valeur propre peut être approchée par la méthode de la puissance, décrire algorithme 2.1. Il s'agit d'une boucle récursive dans laquelle, à partir d'un résidu du produit d'une matrice diagonalisable par un vecteur initiale quelconque, chaque itération consiste à multiplier la matrice par le résidu courant et à mettre à jour le résidu normalisé. Cette suite tend vers la plus grande valeur propre de la matrice.

Remarque 2.11 Dans le code DIVA, nous mesurons l'énergie discrète suivant la formule (2.104). Nous avons également modifié la CFL en implémentant la méthode de la puissance, qui a l'avantage de s'appliquer indépendamment à des configurations isotropes et anisotropes.

2.3.2 Courbes d'énergies des études isotrope et VTI 2D

Mesurer l'énergie discrète au cours d'une simulation permet également d'évaluer les réflexions parasites dues aux CLA. L'énergie discrète, tout comme l'énergie continue, s'exprime en J.s^{-2} car nous travaillons en vitesse et non en déplacement (et en considérant la masse volumique en 2D comme une masse surfacique).

Algorithme 2.1 : Méthode de la puissance

Data : $A, \mathbf{u}_0, \epsilon$
Result : λ_{max}
1 $\mathbf{r} \leftarrow A\mathbf{u}_0$;
2 $\lambda_{max} \leftarrow \langle \mathbf{r}, \mathbf{u}_0 \rangle / \langle \mathbf{u}_0, \mathbf{u}_0 \rangle$;
3 **do**
4 $\lambda_{tmp} \leftarrow \lambda_{max}$;
5 $\mathbf{r}_{tmp} \leftarrow A\mathbf{r}$;
6 $\lambda_{max} \leftarrow \langle \mathbf{r}_{tmp}, \mathbf{r} \rangle / \langle \mathbf{r}, \mathbf{r} \rangle$;
7 $\mathbf{r} \leftarrow \mathbf{r}_{tmp}$;
8 **while** ($|\lambda_{max} - \lambda_{tmp}| > \epsilon$) ;

Cas isotrope

La Fig. 2.12 décrit le comportement de l'énergie discrète pour des sources données par une onde P, une onde S et une source PS. Ces expériences se rapportent aux images des fronts d'ondes présentées Fig. 2.2, 2.3 et 2.4. Le cas homogène permet de décrypter les différentes étapes de la simulation. L'énergie croît fortement puis décroît par paliers, le pic correspond à l'émission de la source puis le premier palier, à $t = 1s$ environ, correspond à la propagation des ondes avant contact avec le bord absorbant. Le second palier, à $t = 5s$ environ, est dû aux réflexions parasites de la CLA. En effet, après le passage des ondes principales, il ne reste temporairement que les réflexions, jusqu'à rencontrer un autre bord absorbant. C'est sur ce second palier que l'efficacité d'une CLA peut être mesurée.

Sur les trois configurations i.e. homogène, bi-couches bande et coin, les différences entre les trois types de source s'expliquent simplement. Une simulation avec une source en onde P ou en onde S a la même énergie initiale tandis qu'une simulation avec une source en onde PS a une énergie initiale égale à la somme des énergies relatives aux deux ondes. Une simulation avec une source en onde P, dont la vitesse de propagation est plus rapide, voit son énergie décroître avant celle d'une simulation avec une source en onde S, car les ondes P atteignent le bord absorbant plus rapidement. Les réflexions pointées par les courbes d'énergies sont plus fortes pour une source en onde S par rapport à une source en onde P, ce qui est dû aux choix d'approximation et de simplification des CLA. Du coup, une simulation avec une source PS a une courbe d'énergie correspondant plutôt à l'absorption de l'onde S et ne permettant pas de distinguer les petites réflexions de l'onde P.

Cas VTI

De même que pour le cas isotrope, la Fig. 2.13 décrit le comportement de l'énergie discrète pour des sources données par une onde P, une onde S et une source PS. Ces expériences se rapportent aux images des fronts d'ondes présentées sur les Fig. 2.6, 2.7 et 2.8. Le cas homogène permet de visualiser les paliers des différentes étapes de la simulation. C'est sur le second palier, à $t = 5s$ environ, que l'efficacité d'une CLA peut être mesurée. Sur les trois configurations i.e. homogène, bi-couches bande et coin, les conclusions sont les mêmes que pour le cas isotrope.

Comparaison avec la CLA isotrope

En utilisant une CLA isotrope dans un milieu VTI, nous avons observé à la Fig. 2.9 que les réflexions d'ondes P étaient très fortes. L'étude de l'énergie sur le cas homogène de cette expérience est présentée sur la Fig. 2.14. Les courbes de la source en onde P confirment les observations, les réflexions de l'onde P transformée en onde P, avec un palier à 0.1, mais aussi en transformée onde S, avec un palier à 0.01, sont plus fortes avec une CLA isotrope qu'avec une CLA VTI où nous observons un palier en dessous de 0.01. En comparaison avec les simulations dotées d'une source en onde PS ou en onde S, la première réflexion (onde P transformée en onde P) se produit alors que l'onde S n'est pas encore totalement absorbée et la deuxième réflexion (onde P transformée en onde S) a une intensité plus faible que les réflexions de l'onde S (transformée en ondes P et S). Ces réflexions sont visibles sur les images des fronts d'ondes mais ne se détectent que sur les courbes d'énergie de la simulation avec une source en onde P.

Non-elliptique

Les énergies correspondant aux expériences avec un milieu VTI non-elliptique (Fig. 2.10) sont présentées sur la Fig. 2.15 pour une source élastique. Le comportement est le même que pour un milieu elliptique (Fig. 2.13) ou isotrope (Fig. 2.12). La comparaison avec le cas homogène pour une CLA isotrope est présentée à la Fig. 2.16, correspondant à l'expérience de la Fig. 2.11 où la CLA isotrope est utilisée dans un milieu VTI non-elliptique. Comme la source est élastique, les réflexions des ondes P visibles sur les images ne sont pas évaluables par les courbes d'énergie.

Simulation en temps long

Nous avons prouvé la stabilité du problème mixte discret modulo une CFL. Néanmoins, nous savons que l'utilisation de conditions aux limites instables peut faire que la solution numérique se mette à croître de façon exponentielle après un temps long de simulation, c'est-à-dire bien après les réflexions des ondes. Nous proposons donc ici de mesurer l'énergie discrète pour un temps long de simulation. La Fig. 2.17 regroupe les énergies discrètes mesurées sur des cas isotrope, VTI elliptique et VTI non-elliptique pour des configurations homogène, bi-couches bande et coin avec une source élastique. Une fois la précision machine atteinte, la mesure de l'énergie ne montre aucune instabilité en temps long.

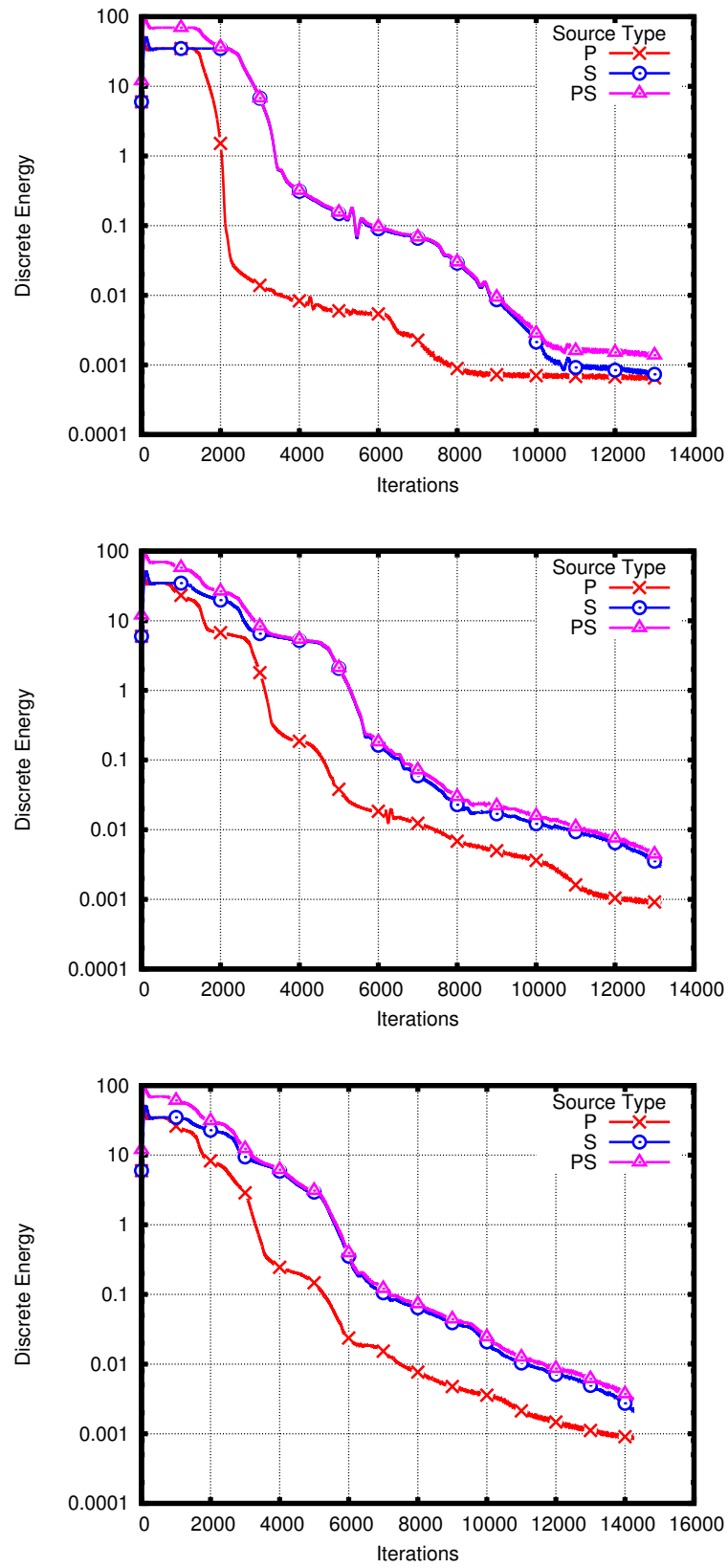


FIGURE 2.12 – Énergie discrète pour les cas isotrope 2D homogène (haut) et bi-couches, bande (centre) et coin (bas)

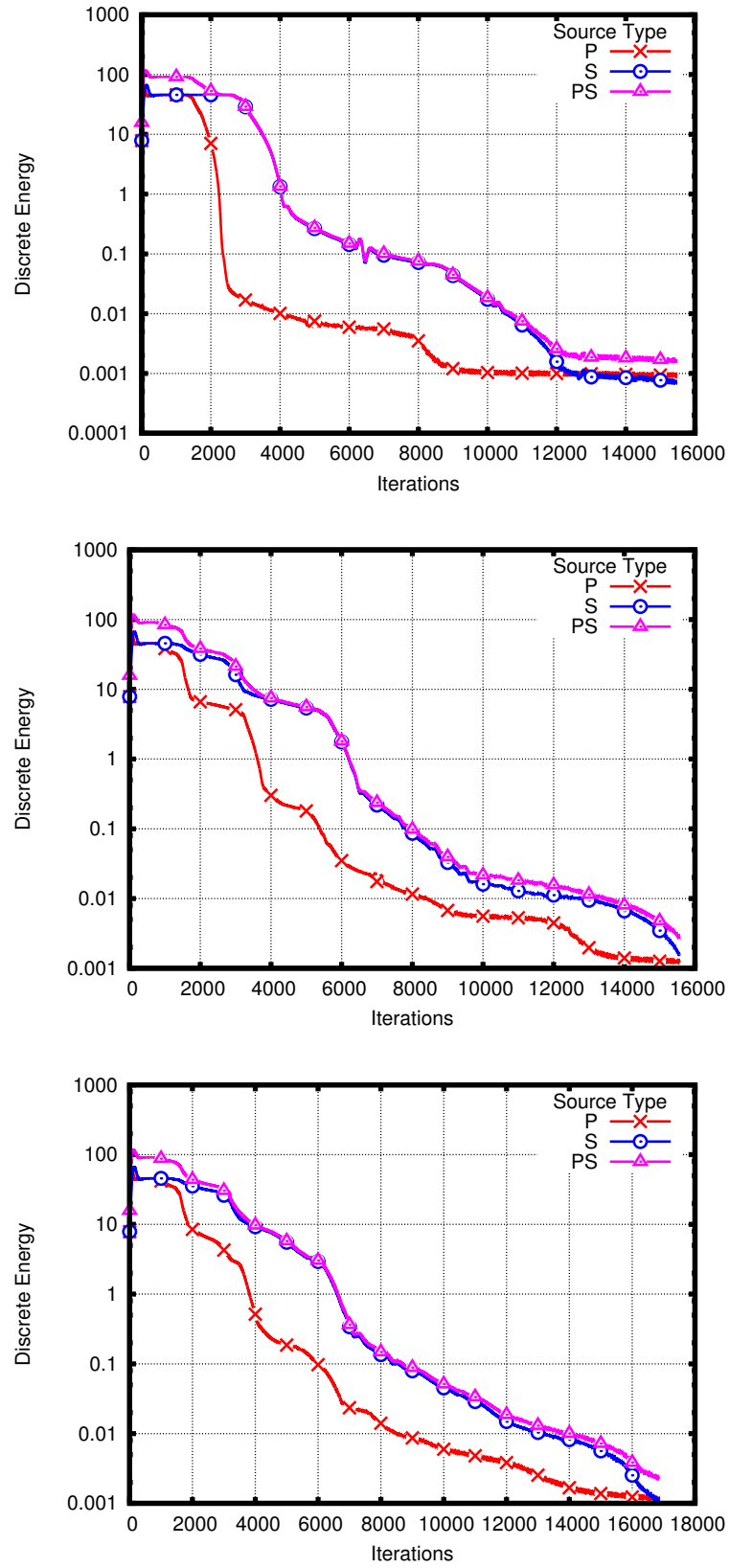


FIGURE 2.13 – Énergie discrète pour les cas VTI 2D homogène (haut) et bi-couches, bande (centre) et coin (bas)

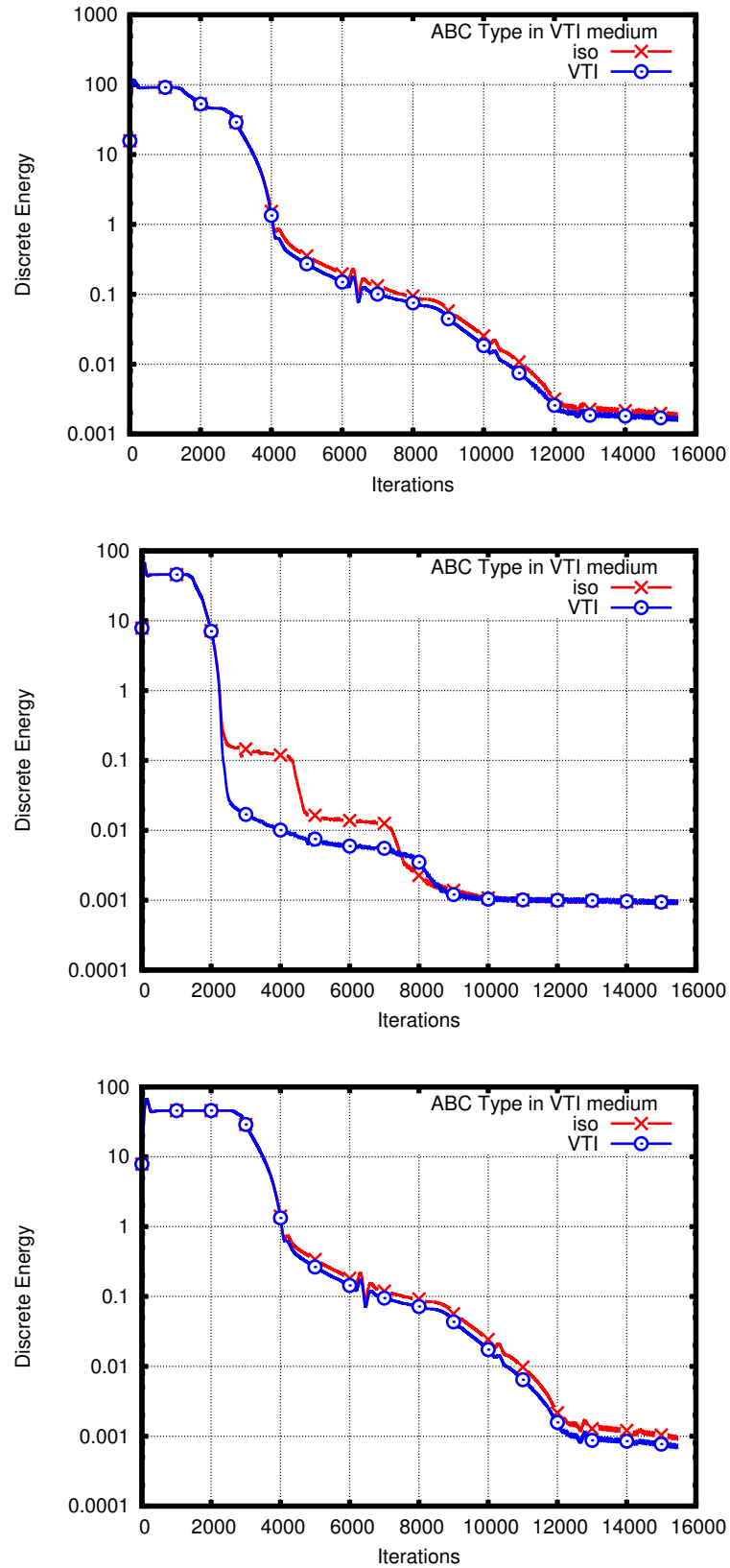


FIGURE 2.14 – Comparaison avec la CLA isotrope dans un milieu homogène VTI 2D, avec une source PS (haut), une source P (centre) et une source S (bas)

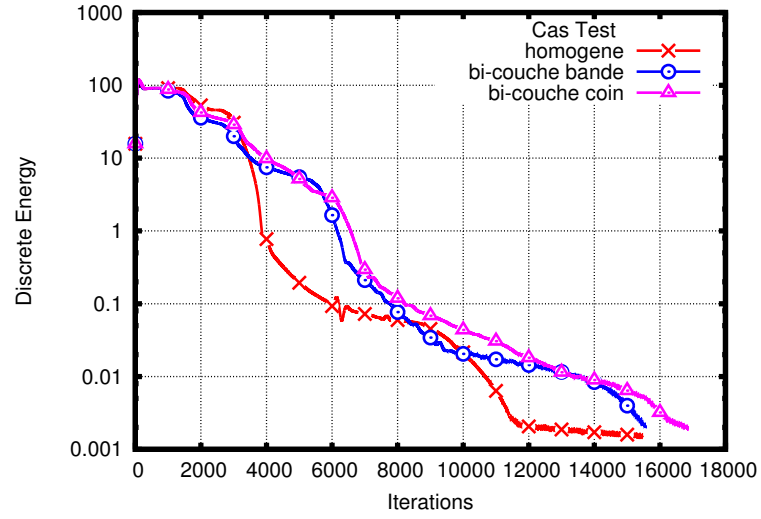


FIGURE 2.15 – Énergie discrète dans les cas VTI 2D non-elliptique

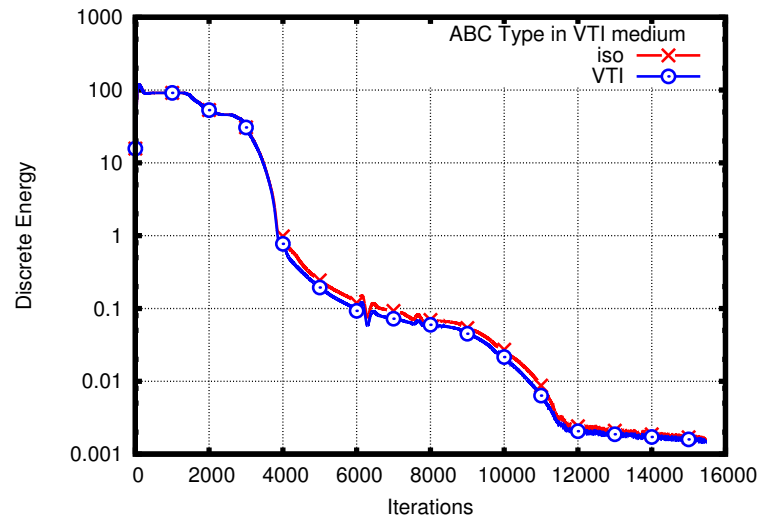


FIGURE 2.16 – Comparaison avec la CLA isotrope sur un cas VTI 2D non-elliptique

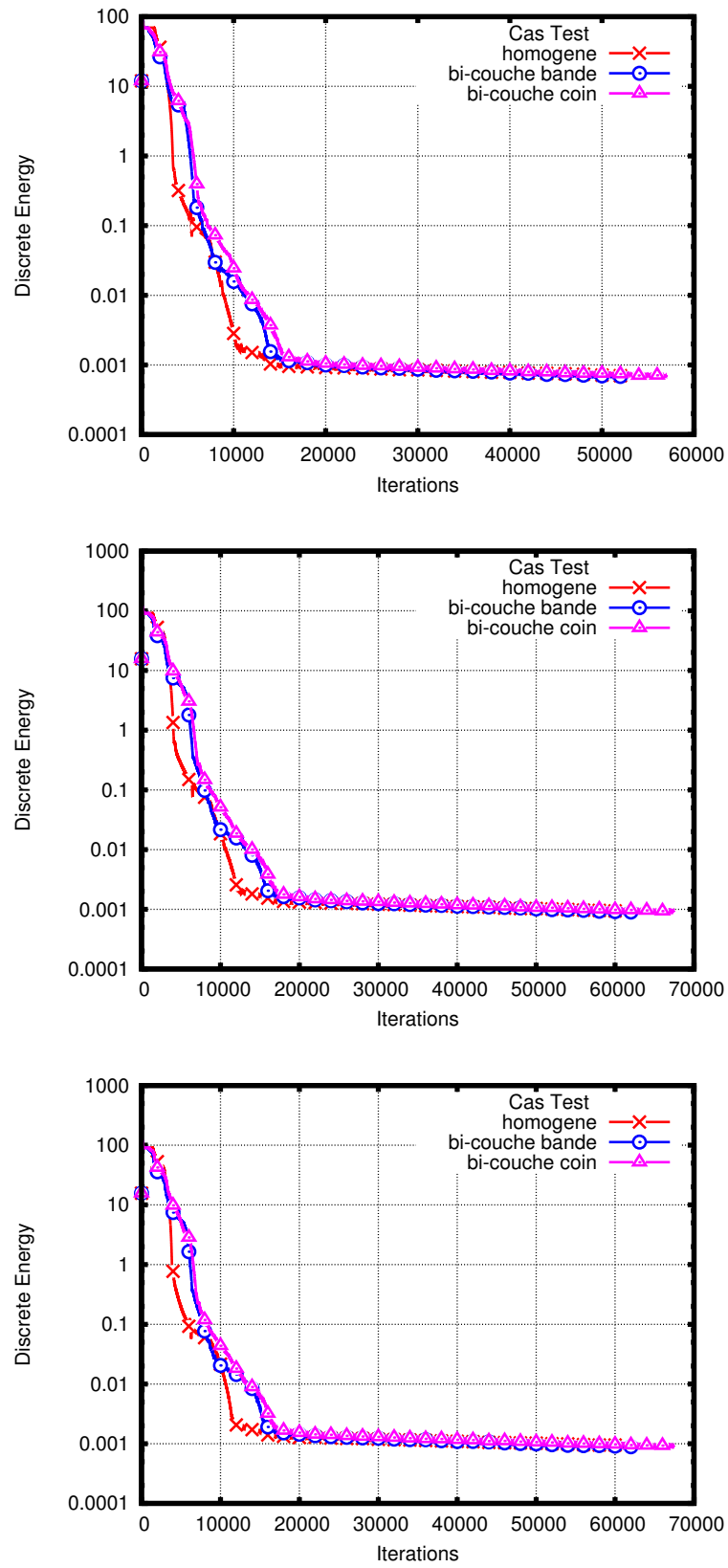


FIGURE 2.17 – Énergie discrète pour des temps long dans les cas isotrope (haut), VTI elliptique (centre) et VTI non-elliptique (bas) 2D

2.4 Coefficients de réflexion de la CLA VTI 2D

Au paragraphe précédent, nous avons étudié la stabilité de la CLA. Celle-ci est démontrée à l'aide d'une technique énergétique. Dans ce paragraphe, nous nous intéressons aux réflexions causées par le bord absorbant. Il s'agit donc d'analyser le comportement de la solution du problème mixte. Les coefficients de réflexion ont déjà été calculés en isotrope, par exemple dans [Tso99] et nous allons suivre la même démarche en anisotrope.

Nous considérons ici un domaine semi-infini $\{\Omega : x \in]-\infty, 0]\}$ avec une CLA en $\{x = 0\}$. Au bord du domaine, la condition aux limites absorbante réfléchit une partie des ondes. Une onde P produit deux réflexions d'ondes notées PP et PS, et une onde S produit deux réflexions d'ondes notées SP et SS. Le champ d'onde total est la somme de l'onde incidente et des ondes réfléchies :

$$\mathbf{u}^I = \mathbf{u}^{inc_I} + \mathbf{u}^{II} + \mathbf{u}^{IJ} \quad (2.111)$$

avec (I, J) représentant (P, S) pour une onde incidente P et (S, P) pour une onde incidente S.

Soit la matrice de l'équation de dispersion du système de l'élastodynamique en dimension deux :

$$E_{VTI,2D} = \begin{pmatrix} C_{11}k_x^2 + C_{44}k_z^2 & (C_{44} + C_{13})k_xk_z \\ (C_{44} + C_{13})k_xk_z & C_{44}k_x^2 + C_{33}k_z^2 \end{pmatrix} \quad (2.112)$$

où C est la matrice d'élasticité VTI 2D (2.31).

Sous l'hypothèse d'ellipticité ($\kappa = \sqrt{1 + 2\varepsilon} = \sqrt{1 + 2\delta}$), les vecteurs propres de la matrice $E_{VTI,2D}$ sont donnés par :

$$\begin{pmatrix} Xik_x \\ Zik_z \end{pmatrix}, \begin{pmatrix} -Zik_z \\ Xik_x \end{pmatrix} \quad (2.113)$$

avec $X = \sqrt{\kappa^2 V_p^2 + V_s^2}$ et $Z = \sqrt{V_p^2 + V_s^2}$.

Nous notons A_{II} et A_{IJ} les coefficients de réflexion. Ils interviennent de la façon suivante :

$$\begin{cases} \mathbf{u}^{inc_P} = \begin{pmatrix} Xik_x \\ Zik_z \end{pmatrix} p^{inc}, \mathbf{u}^{PP} = \begin{pmatrix} Xik_x^I \\ Zik_z^I \end{pmatrix} A_{PPP}^P, \mathbf{u}^{PS} = \begin{pmatrix} -Zik_z \\ Xik_x^J \end{pmatrix} A_{PSP}^S \\ \mathbf{u}^{inc_S} = \begin{pmatrix} -Zik_z \\ Xik_x \end{pmatrix} p^{inc}, \mathbf{u}^{SS} = \begin{pmatrix} -Zik_z \\ Xik_x^S \end{pmatrix} A_{SSP}^S, \mathbf{u}^{SP} = \begin{pmatrix} -Xik_x^P \\ -Zik_z \end{pmatrix} A_{SPP}^P \end{cases} \quad (2.114)$$

avec

$$p^{inc} = e^{i\omega t + ik_x x + ik_z z}, p^I = e^{i\omega t + ik_x^I x + ik_z^I z}, p^J = e^{i\omega t + ik_x^J x + ik_z^J z} \quad (2.115)$$

pour un bord de normale extérieure colinéaire à \mathbf{e}_x , sur lequel nous avons $k_z^J = k_z^I = k_z$.

Les coefficients de réflexion s'expriment en fonction d'un paramètre θ_0 qui correspond à l'angle d'incidence de l'onde sur le bord absorbant. Ce paramètre détermine la valeur des rapports $\frac{k_z}{w}$. Dans le cas VTI, la Fig. 2.18 représente ces rapports pour une onde incidente P ou S. Les rapports $\frac{k_z}{w}$ et $\frac{k_x}{w}$ se déterminent facilement par trigonométrie. Les rapports $\frac{k_x^I}{w}$ et

$\frac{k_x^J}{w}$ se déduisent des premiers, l'un par symétrie, l'autre par la loi de Snell-Descartes

$$\begin{cases} \frac{\sin \theta_1}{V_s} = \frac{\sin \theta_0}{V_p}, \text{ pour l'onde P,} \\ \frac{\sin \theta_1}{V_p} = \frac{\sin \theta_0}{V_s}, \text{ pour l'onde S.} \end{cases} \quad (2.116)$$

Ainsi, il est possible d'écrire :

$$\begin{cases} \frac{k_x}{w} = \frac{\cos \theta_0}{\kappa V_p}, \frac{k_z}{w} = \frac{\sin \theta_0}{V_p}, \frac{k_x^P}{w} = -\frac{\cos \theta_0}{\kappa V_p}, \frac{k_x^S}{w} = -\sqrt{\frac{1}{V_s^2} - \frac{\sin^2 \theta_0}{V_p^2}}, \text{ pour une onde P} \\ \frac{k_x}{w} = \frac{\cos \theta_0}{V_s}, \frac{k_z}{w} = \frac{\sin \theta_0}{V_s}, \frac{k_x^S}{w} = -\frac{\cos \theta_0}{V_s}, \frac{k_x^P}{w} = -\sqrt{\frac{1}{\kappa^2 V_p^2} - \frac{\sin^2 \theta_0}{V_s^2}}, \text{ pour une onde S} \end{cases} \quad (2.117)$$

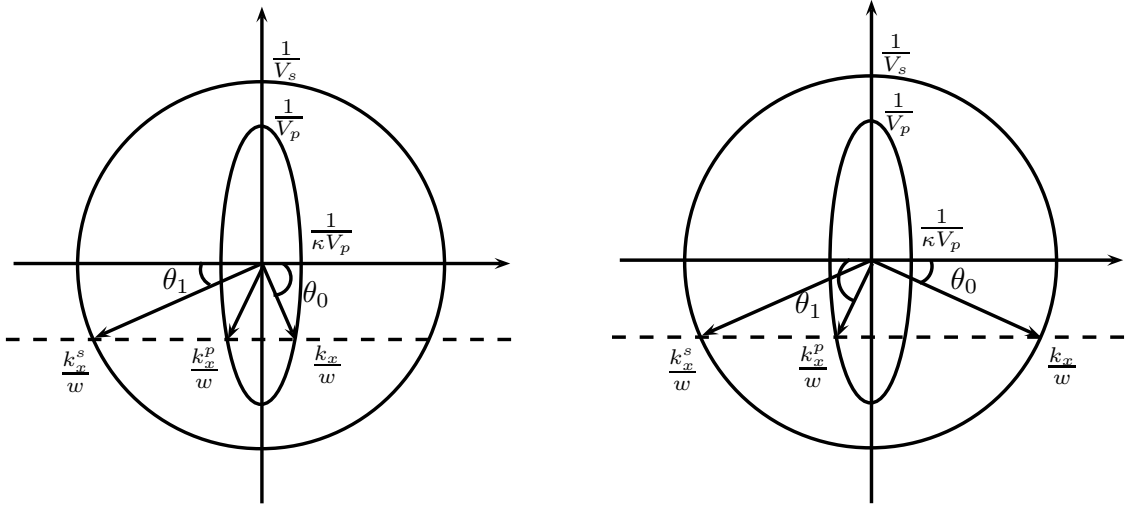


FIGURE 2.18 – Schéma des angles intervenant dans la réflexion de l'onde P (gauche) et de l'onde S (droite), en VTI

Remarque 2.12 *Au-delà d'un certain angle critique, et comme physiquement $V_s < V_p$, le terme sous la racine carrée de la deuxième équation du système (2.117) devient négatif, ce qui conduit à un rapport $\frac{k_x^P}{w}$ complexe pour l'onde S. Nous retrouvons également cette propriété sur la Fig. 2.18.*

Sur un bord de normale extérieure colinéaire à \mathbf{e}_x , le champ total \mathbf{u} vérifie la CLA (2.53) :

$$\begin{cases} C_{11}\partial_x u_x + C_{13}\partial_z u_z &= -\rho \kappa V_p \partial_t u_x \\ C_{13}\partial_x u_x + C_{33}\partial_z u_z &= 0 \\ C_{44}\partial_x u_z + C_{44}\partial_z u_x &= -\rho V_s \partial_t u_z \end{cases} \quad (2.118)$$

Chapitre 2. Conditions aux limites absorbantes pour des matériaux VTI

En utilisant (2.114) dans la première et la dernière équation de (2.118), il vient par exemple pour une onde incidente P :

$$\left\{ \begin{array}{l} C_{11}[-Xk_x^2 p^{inc} - Xk_x^p A_{pp} p^{rp} - Zk_x^s k_z A_{ps} p^{rs}] \\ \quad + C_{13}[-Zk_z^2 p^{inc} - Zk_z^2 A_{pp} p^{rp} + Xk_x^s k_z A_{ps} p^{rs}] \\ \quad = \rho \kappa V_p [-Xk_x w p^{inc} + Xk_x^p A_{pp} w p^{rp} + Zk_z w A_{ps} p^{rs}] \\ C_{44}[-Zk_x k_z p^{inc} + Zk_x^p k_z A_{pp} p^{rp} - Xk_x^s A_{ps} p^{rs}] \\ \quad + C_{44}[-Xk_x k_z p^{inc} + Xk_x^p k_z A_{pp} p^{rp} + Zk_z^2 A_{ps} p^{rs}] \\ \quad = \rho V_s [-Zk_z w p^{inc} - Zk_z w A_{pp} p^{rp} + Xk_x^s w A_{ps} p^{rs}] \end{array} \right. \quad (2.119)$$

En utilisant le fait que sur la frontière, c'est-à-dire en $\{x = 0\}$, les exponentielles (2.115) des termes p^\star sont toutes égales, il en résulte un système matriciel définissant A_{PP} et A_{SP} :

$$\begin{pmatrix} -XC_{11}k_x^2 - ZC_{13}k_z^2 - X\kappa k_x^p w \rho V_p & (XC_{13} - ZC_{11})k_x^s k_z - Z\kappa k_z w \rho V_p \\ (X + Z)C_{44}k_x^p k_z + Zk_z w \rho V_s & -XC_{44}k_x^s k_z^2 + ZC_{44}k_z^2 - Xk_x^s w \rho V_s \end{pmatrix} \begin{pmatrix} A_{PP} \\ A_{PS} \end{pmatrix} = \begin{pmatrix} XC_{11}k_x^2 + ZC_{13}k_z^2 - X\kappa k_x w \rho V_p \\ (X + Z)C_{44}k_x k_z - Zk_z w \rho V_s \end{pmatrix} \quad (2.120)$$

Un développement limité peut être effectué à l'aide du logiciel MapleTM, voir annexe A.4, sur les coefficients de réflexion, autour de l'angle d'incidence normale $\theta_0 = 0$:

$$\left\{ \begin{array}{l} A_{PP} = \frac{\kappa^2 V_p^2 V_s^2 - \left(\sqrt{V_p^2 - V_s^2} \sqrt{\kappa^2 V_p^2 - V_s^2} - V_s^2 \right)^2}{4(\kappa V_p - V_s) V_p^3 \kappa} \theta_0^2 + O(\theta_0^4) \\ A_{PS} = \frac{\left(\sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} - V_s(\kappa V_p + V_s) \right) V_s}{2(\kappa V_p + V_s) V_p^2 \kappa} \theta_0 + O(\theta_0^3) \\ A_{SS} = \left(\frac{\kappa^2 V_p^2 V_s^2 - \left(\sqrt{V_p^2 - V_s^2} \sqrt{\kappa^2 V_p^2 - V_s^2} - V_s^2 \right)^2}{4(\kappa V_p - V_s) V_p V_s^2 \kappa} + \frac{V_p^2 - V_s^2}{4V_s^2} \right) \theta_0^2 + O(\theta_0^4) \\ A_{SP} = \frac{\sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} - V_s(\kappa V_p + V_s)}{2(\kappa V_p + V_s) V_s} \theta_0 + O(\theta_0^3) \end{array} \right. \quad (2.121)$$

Remarque 2.13 Lorsque $\kappa = 1$, les coefficients de réflexion correspondent au cas isotrope. Dans ce cas leurs modules sont comparables à ceux obtenus dans [Tso99] excepté pour A_{SP} et A_{PS} qui diffèrent d'un facteur $\frac{V_p}{V_s}$.

La Fig. 2.19 représente ces coefficients de réflexion pour un milieu VTI elliptique, avec la CLA VTI et la CLA isotrope, ainsi que pour un milieu isotrope (avec une CLA isotrope). Les courbes sont cohérentes et comparables à la référence isotrope. À incidence normale il n'y a aucune réflexion, à incidence rasante les coefficients A_{PP} et A_{SS} génèrent une réflexion totale. Le coefficient A_{PS} est assez faible tandis que le coefficient A_{SP} est plus important. De

plus, A_{PS} et A_{SP} sont nuls à incidence normale et parfaitement rasante et présentent un pic à incidence presque rasante. Enfin, tous les coefficients sont petits à incidence presque normale.

La comparaison entre l'utilisation d'une CLA VTI et d'une CLA isotrope dans un milieu VTI met en évidence la supériorité de la CLA VTI à incidence proche de la normale. Ceci s'observe par des coefficients moins élevés et surtout par le coefficient A_{PP} de la CLA isotrope qui est différent de zéro à $\theta_0 = 0$ (il est égal à $\frac{\kappa-1}{\kappa+1}$). Les choix d'approximation et de simplification ont été faits dans ce sens. Par contre, à incidence presque rasante, c'est la CLA isotrope qui génère des réflexions moins importantes.

Remarque 2.14 *Ces résultats sont à corrélérer avec les images des fronts d'ondes de la CLA VTI (Fig. 2.6), de la CLA isotrope dans un milieu VTI (Fig. 2.9) et des courbes d'énergies discrètes (Fig. 2.14).*

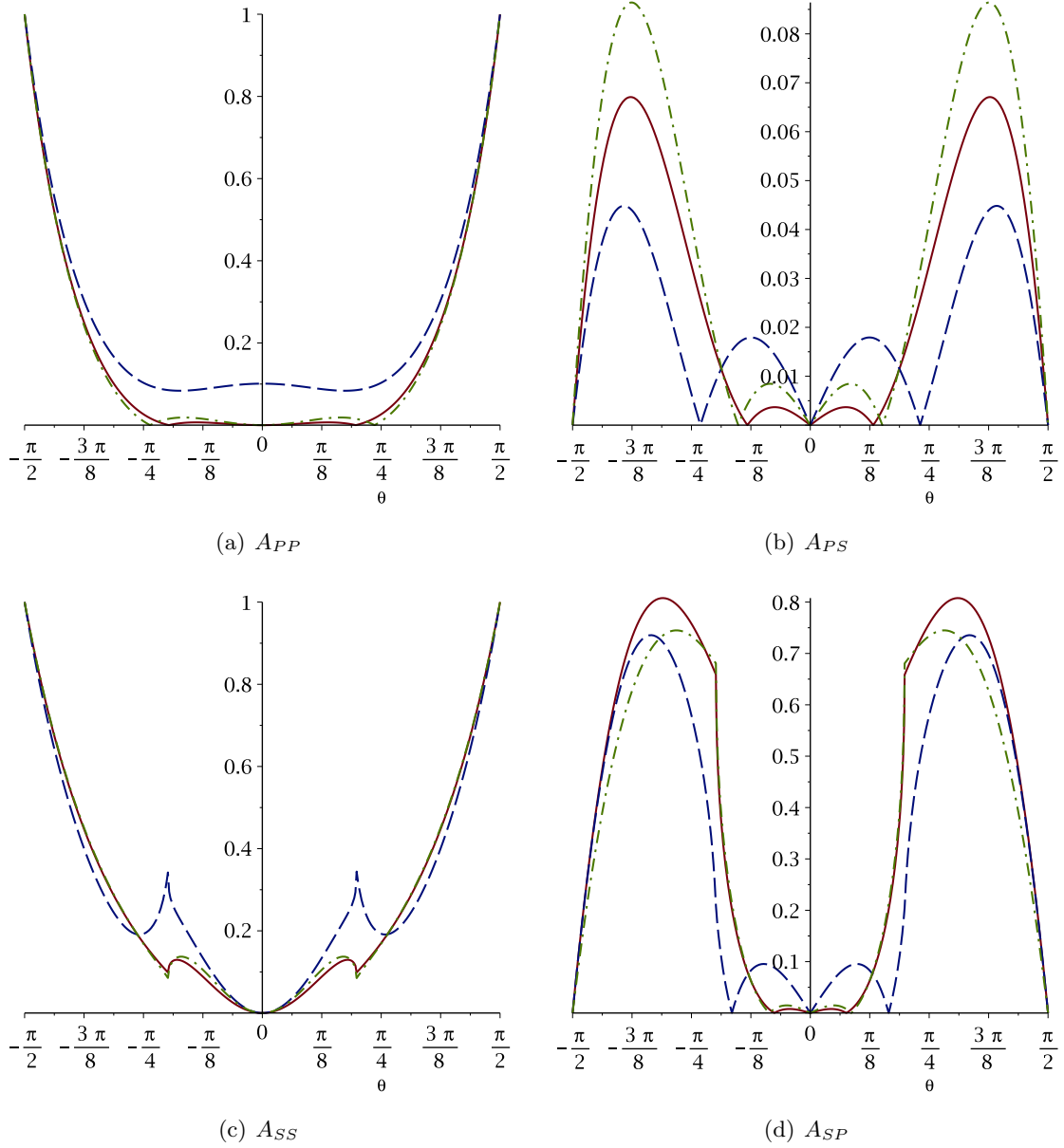


FIGURE 2.19 – Coefficients de réflexion de la CLA VTI 2D pour $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1750 \text{ m.s}^{-1}$ et $\varepsilon = 0.25$, en fonction de θ_0 , en rouge trait plein, comparaison avec l'utilisation d'une CLA isotrope dans le même milieu VTI elliptique, en bleu trait pointillé; référence de la configuration isotrope (CLA et milieu), en vert trait semi-pointillé

2.5 Conclusion

Nous avons donc mis en œuvre une technique de construction de CLA basée sur la diagonalisation du système de l'élastodynamique. Nous avons retrouvé les CLA connues pour des matériaux isotropes et orthotropes 2D et nous avons pu en déduire une CLA pour des matériaux VTI 2D. Cette condition aux limites peut s'interpréter comme la combinaison des

deux CLA obtenues en limitant les calculs aux cas pseudo-acoustiques des ondes P seules et ondes S seules. Nous pouvons aussi remarquer que la condition aux limites ne dépend pas du coefficient δ de Thomsen. C'est certainement dû au fait que la condition correspond à l'approximation d'ordre le plus bas d'une condition transparente.

Afin de mieux évaluer les réflexions dues aux CLA proposées, nous avons construit des sources et des conditions initiales permettant de ne générer qu'une onde P ou qu'une onde S dans un matériau VTI elliptique. L'étude expérimentale a révélé un comportement tout à fait similaire dans des configurations isotropes et VTI 2D (elliptique ou non), tant sur la localisation que sur l'intensité des réflexions parasites. En comparant ensuite la CLA VTI et la CLA isotrope cette fois dans un même milieu VTI 2D (elliptique ou non), nous avons illustré la supériorité de la CLA VTI, essentiellement sur la composante en onde P, la composante en onde S étant strictement la même dans le cas elliptique.

Nous avons ensuite étudié la stabilité du problème mixte continu résultant du couplage de l'élastodynamique avec la CLA VTI 2D et du problème semi-discret associé écrit en formulation DG avec flux centrés. Pour cela, nous avons opté pour une méthode basée sur l'énergie associée au problème. La stabilité de l'énergie continue est obtenue formellement grâce à la positivité des paramètres physiques intervenant dans les équations. Pour étendre cette propriété au problème discret, nous avons choisi de coupler la formulation DG à flux centrés avec un schéma en temps de type Leap-Frog et nous démontrons que nous pouvons définir une énergie discrète, c'est-à-dire une fonctionnelle positive à chaque pas de temps, sous une condition de type CFL. La stabilité du problème discret découle alors de la décroissance de l'énergie discrète. Les résultats de stabilité sont illustrés par des courbes d'énergies qui montrent aussi que la CLA VTI fonctionne mieux que la CLA isotrope dans un milieu VTI 2D (elliptique ou non).

Pour finir, nous avons calculé les coefficients de réflexion A_{PP} , A_{PS} , A_{SP} et A_{SS} de la CLA VTI en fonction de l'angle d'incidence θ_0 pour un milieu VTI elliptique. Nous avons vérifié que ces quatre coefficients étaient bien nuls à incidence normale (i.e. lorsque $\theta_0 = 0$). En utilisant des développements limités au voisinage de $\theta_0 = 0$, nous avons montré que A_{PP} et A_{SS} étaient équivalents à θ_0^2 et que A_{PS} et A_{SP} étaient équivalents à θ_0 . Nous retrouvons donc le même comportement que dans le cas d'une CLA isotrope appliquée à un milieu isotrope. De plus, nous avons montré que la CLA VTI était meilleure que la CLA isotrope dans un milieu VTI elliptique. En particulier, le coefficient A_{PP} de la CLA isotrope n'est plus nul en $\theta_0 = 0$ dans ce cas mais égal à $\frac{\kappa-1}{\kappa+1}$. Ainsi, moins le milieu est isotrope, moins la CLA isotrope est efficace.

En résumé, nous avons obtenu les résultats suivants :

Théorème 2.1 *Une CLA pour des matériaux VTI 2D dans le cas d'une frontière plane de normale extérieure \mathbf{e}_x est :*

$$\begin{cases} \sigma_{xx} &= -\rho V_p \sqrt{1 + 2\varepsilon} v_x \\ \sigma_{xz} &= -\rho V_s v_z \end{cases} \quad (2.122)$$

Théorème 2.2 *Pour ne générer que des ondes P ou des ondes S dans un matériau VTI elliptique, il suffit d'appliquer au système de l'élastodynamique non homogène :*

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) + s_R(t) \underline{\underline{s}}(\mathbf{x}) \end{cases} \quad (2.123)$$

les sources :

$$\begin{cases} \underline{s}(\mathbf{x}) = \delta_{x_0}(\mathbf{x}) \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \\ 0 & \sqrt{V_p^2 - V_s^2} \end{pmatrix}, & \text{pour ne g n rer qu'une onde P,} \\ \underline{s}(\mathbf{x}) = \delta_{x_0}(\mathbf{x}) \begin{pmatrix} 0 & -\sqrt{V_p^2 - V_s^2} \\ \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \end{pmatrix}, & \text{pour ne g n rer qu'une onde S,} \end{cases} \quad (2.124)$$

o , par exemple, s_R est donn e comme une ondelette de Ricker de fr quence pic f_p :

$$s_R(t) = (1 - 2\pi^2 f_p^2 t^2) e^{-\pi^2 f_p^2 t^2} \quad (2.125)$$

ou de consid rer le probl me de Cauchy :

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\sigma}(\mathbf{x}, t) \\ \partial_t \underline{\sigma}(\mathbf{x}, t) &= \underline{C}(\mathbf{x}) \underline{\epsilon}(\mathbf{v}(\mathbf{x}, t)) \\ \mathbf{v}(0, \mathbf{x}) &= \mathbf{v}_0(\mathbf{x}) \\ \underline{\sigma}(0, \mathbf{x}) &= \underline{\sigma}_0(\mathbf{x}) \end{cases} \quad (2.126)$$

en choisissant pour conditions initiales $\underline{\sigma}_0 = 0$ et :

$$\begin{cases} \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} (x - x_0) \\ \sqrt{V_p^2 - V_s^2} (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, & \text{pour ne g n rer qu'une onde P,} \\ \mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} -\sqrt{V_p^2 - V_s^2} (z - z_0) \\ \sqrt{\kappa^2 V_p^2 - V_s^2} (x - x_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, & \text{pour ne g n rer qu'une onde S,} \end{cases} \quad (2.127)$$

Th or me 2.3 Les probl mes mixtes continu, semi-discret et discret obtenus en couplant l' lastodynamique avec la CLA VTI 2D (2.122) sont stables au sens suivant :

– pour le probl me continu, nous d finissons la forme quadratique :

$$\mathbb{E}(t) = \frac{1}{2} \left(\|\mathbf{v}\|_{\Omega, \rho}^2 + \|\underline{\sigma}\|_{\Omega, \underline{C}^{-1}}^2 \right) \quad (2.128)$$

– pour le probl me semi-discret, nous d finissons la forme :

$$\mathbb{E}_{semi_d}(t) = \langle \partial_t M_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h \rangle + \langle \partial_t M_{\underline{\sigma}, C^{-1}} \underline{\sigma}_h, \underline{\sigma}_h \rangle \quad (2.129)$$

– pour le probl me mixte discret, nous d finissons la forme :

$$\mathbb{E}_d^n = \frac{1}{2} \langle M_{\mathbf{v}} \mathbf{v}_h^n, \mathbf{v}_h^n \rangle + \frac{1}{2} \langle M_{\underline{\sigma}, C^{-1}} \underline{\sigma}_h^{n+\frac{1}{2}}, \underline{\sigma}_h^{n-\frac{1}{2}} \rangle \quad (2.130)$$

Alors, (2.128) et (2.129) sont toujours décroissantes au cours du temps et (2.130) est positive et décroissante si et seulement si la matrice

$$M_v - \frac{\Delta t^2}{4} R_{\sigma} (M_{\sigma, C})^{-1} (R_{\sigma})^T \quad (2.131)$$

est positive.

Corollaire 2.3.1 *Le problème mixte discret est stable sous la condition de type CFL*

$$1 - \frac{\Delta t^2}{4} \lambda_{max} \geq 0 \quad (2.132)$$

où λ_{max} est la plus grande valeur propre de la matrice $M_v^{-1} R_{\sigma} (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T$.

Théorème 2.4 *Les coefficients de réflexion de la CLA VTI 2D, autour de l'angle d'incidence normale $\theta_0 = 0$, sont donnés par :*

$$\left\{ \begin{array}{l} A_{PP} = \frac{\kappa^2 V_p^2 V_s^2 - \left(\sqrt{V_p^2 - V_s^2} \sqrt{\kappa^2 V_p^2 - V_s^2} - V_s^2 \right)^2}{4(\kappa V_p - V_s) V_p^3 \kappa} \theta_0^2 + O(\theta_0^4) \\ A_{PS} = \frac{\left(\sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} - V_s(\kappa V_p + V_s) \right) V_s}{2(\kappa V_p + V_s) V_p^2 \kappa} \theta_0 + O(\theta_0^3) \\ A_{SS} = \left(\frac{\kappa^2 V_p^2 V_s^2 - \left(\sqrt{V_p^2 - V_s^2} \sqrt{\kappa^2 V_p^2 - V_s^2} - V_s^2 \right)^2}{4(\kappa V_p - V_s) V_p V_s^2 \kappa} + \frac{V_p^2 - V_s^2}{4V_s^2} \right) \theta_0^2 + O(\theta_0^4) \\ A_{SP} = \frac{\sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} - V_s(\kappa V_p + V_s)}{2(\kappa V_p + V_s) V_s} \theta_0 + O(\theta_0^3) \end{array} \right. \quad (2.133)$$

De plus, l'utilisation d'une CLA isotrope dans un milieu VTI elliptique modifie les facteurs des coefficients et ajoute un terme en $o(1)$ égal à $\frac{\kappa-1}{\kappa+1}$ sur le coefficient A_{PP} .

Chapitre 3

Conditions aux limites absorbantes pour des matériaux TTI

Ce chapitre propose une solution au problème général de la formulation de conditions aux limites pour des matériaux TTI. Nous commençons par illustrer la complexité de la méthodologie d'Engquist et Majda qui intervient dès la 2D. Les difficultés techniques auxquelles nous sommes confrontés nous conduisent à proposer une nouvelle méthode permettant de traiter le cas TTI elliptique. Il s'agit d'une hypothèse qui peut sembler limitative puisque notre approche ne permet pas de traiter tous les cas de matériaux anisotropes. Toutefois, nous pouvons expliquer pourquoi l'hypothèse d'ellipticité ne l'est pas forcément. Le processus de construction repose sur l'utilisation de repères géométriques par les courbes de lenteur et sur un changement de variables du cas isotrope vers le cas TTI. Nous nous focalisons ici sur les ondes P, de première importance pour les géophysiciens et après avoir vérifié que les ondes S peuvent se traiter comme dans un milieu isotrope. Nous traitons ensuite le cas de la 3D en utilisant exactement la même technique.

3.1 Processus de construction en 2D

3.1.1 Remarques préliminaires

Comme au chapitre précédent, nous nous plaçons en dimension deux. Nous conservons la notation (x, z) . La matrice d'élasticité (1.26) devient :

$$C_{TTI,2D} = \begin{bmatrix} C_{11} & C_{13} & C_{15} \\ & C_{33} & C_{35} \\ & & C_{55} \end{bmatrix} \quad (3.1)$$

Nous rappelons que le système de l'élastodynamique au premier ordre est donné par :

$$\begin{cases} \rho \partial_t v_x &= \partial_x \sigma_{xx} + \partial_z \sigma_{xz} \\ \rho \partial_t v_z &= \partial_x \sigma_{xz} + \partial_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= C_{11} \partial_x v_x + C_{13} \partial_z v_z + C_{15} (\partial_x v_z + \partial_z v_x) \\ \partial_t \sigma_{zz} &= C_{13} \partial_x v_x + C_{33} \partial_z v_z + C_{35} (\partial_x v_z + \partial_z v_x) \\ \partial_t \sigma_{xz} &= C_{15} \partial_x v_x + C_{35} \partial_z v_z + C_{55} (\partial_x v_z + \partial_z v_x) \end{cases} \quad (3.2)$$

et le système au deuxième ordre a pour expression :

$$\begin{cases} \rho \partial_t^2 v_x &= C_{11} \partial_x^2 v_x + (C_{13} + C_{55}) \partial_x \partial_z v_z + C_{55} \partial_z^2 v_x + C_{15} (\partial_x^2 v_z + 2 \partial_x \partial_z v_x) + C_{35} \partial_z^2 v_z \\ \rho \partial_t^2 v_z &= C_{55} \partial_x^2 v_z + (C_{13} + C_{55}) \partial_x \partial_z v_x + C_{33} \partial_z^2 v_z + C_{35} (2 \partial_x \partial_z v_x + \partial_x^2 v_z) + C_{15} \partial_x^2 v_x \end{cases} \quad (3.3)$$

Pour suivre la méthodologie d'Engquist et Majda [EM77, EM79], nous devons travailler dans le domaine de Fourier après avoir gelé la variable normale à la surface sur laquelle la CLA est posée. Pour la frontière $\{x = 0\}$ de normale extérieure \mathbf{e}_x , nous notons (ω, ξ) les variables de Fourier associées à (t, z) . Le système (3.3) se réécrit alors :

$$\begin{cases} C_{11} \partial_x^2 \hat{v}_x - i\xi (C_{13} + C_{55}) \partial_x \hat{v}_z + (\rho\omega^2 - C_{55}\xi^2) \hat{v}_x + C_{15} (\partial_x^2 \hat{v}_z - 2i\xi \partial_x \hat{v}_x) - \xi^2 C_{35} \hat{v}_z &= 0 \\ C_{55} \partial_x^2 \hat{v}_z - i\xi (C_{13} + C_{55}) \partial_x \hat{v}_x + (\rho\omega^2 - C_{33}\xi^2) \hat{v}_z - C_{35} (2i\xi \partial_x \hat{v}_x + \xi^2 \hat{v}_x) + C_{15} \partial_x^2 \hat{v}_x &= 0 \end{cases} \quad (3.4)$$

Comme dans un matériau isotrope (voir section 2.1) ou en VTI (voir section 2.2), nous cherchons à exprimer (3.4) comme une équation différentielle ordinaire de la forme $\partial_x W + MW = 0$. Il n'est pas possible de le faire directement en posant $W = (\hat{v}_x, \hat{v}_z, \partial_x \hat{v}_x, \partial_x \hat{v}_z)$ à cause des termes supplémentaires issus des caractéristiques TTI du milieu. Mais nous pouvons y arriver en introduisant des variables auxiliaires comme par exemple :

$$\begin{bmatrix} \hat{w}_x \\ \hat{w}_z \end{bmatrix} = \begin{bmatrix} C_{11} \partial_x \hat{v}_x + C_{15} \partial_x \hat{v}_z \\ C_{15} \partial_x \hat{v}_x + C_{55} \partial_x \hat{v}_z \end{bmatrix} \quad (3.5)$$

Dans ce cas, W est solution de $\partial_x W + MW = 0$ avec $W = (\hat{w}_x, \hat{w}_z, \hat{v}_x, \hat{v}_z)$ et

$$M = \begin{bmatrix} \frac{-2i\xi C_{15} C_{55} + i\xi C_{15} (C_{13} + C_{55})}{C_{11} C_{55} - C_{15}^2} & \frac{2i\xi C_{15}^2 - i\xi C_{11} (C_{13} + C_{55})}{C_{11} C_{55} - C_{15}^2} & \rho\omega^2 - \xi^2 C_{55} & -\xi^2 C_{35} \\ \frac{2i\xi C_{35} C_{15} - i\xi C_{55} (C_{13} + C_{55})}{C_{11} C_{55} - C_{15}^2} & \frac{-2i\xi C_{35} C_{11} + i\xi C_{15} (C_{13} + C_{55})}{C_{11} C_{55} - C_{15}^2} & -\xi^2 C_{35} & \rho\omega^2 - \xi^2 C_{33} \\ \frac{-C_{55}}{C_{11} C_{55} - C_{15}^2} & \frac{C_{15}}{C_{11} C_{55} - C_{15}^2} & 0 & 0 \\ \frac{C_{15}}{C_{11} C_{55} - C_{15}^2} & \frac{-C_{11}}{C_{11} C_{55} - C_{15}^2} & 0 & 0 \end{bmatrix}$$

Dans l'idée de suivre la même démarche que dans les autres cas, nous avons commencé par chercher les valeurs propres de la matrice M . Nous avons essayé de les calculer en utilisant le logiciel MapleTM sur des jeux d'inconnues auxiliaires et à chaque fois, nous n'avons pas obtenu de valeurs propres faciles à approcher. Devant les difficultés rencontrées dès le cas 2D, nous avons décidé d'abandonner cette piste et de revenir à une compréhension des équations plus basique et fondée sur ce que nous savons faire dans le cas isotrope. Notre objectif est donc désormais d'exploiter les liens explicites entre le cas isotrope et le cas TTI. Nous espérons ainsi pouvoir étendre ce que nous savons faire sur les ondes isotropes aux ondes anisotropes.

3.1.2 Construction d'une CLA pour un matériau TTI 2D

Le système de l'élastodynamique TTI (3.2) pose des difficultés algébriques essentiellement dues aux termes supplémentaires qui apparaissent par rapport au cas isotrope (2.2) ou au

cas VTI (2.32). Mais si nous ramenons la définition d'un milieu à des caractéristiques plutôt géométriques, l'anisotropie TTI n'est qu'une rotation de l'anisotropie VTI, qui n'est elle-même qu'une transformation simple de l'isotrope. Cette remarque est le point-clé sur lequel s'appuie notre méthode de construction de la CLA TTI.

En observant que la CLA d'ordre faible VTI (2.53) ne dépend pas du coefficient de Thomsen δ et comme un milieu TTI n'est qu'une rotation d'un milieu VTI, nous faisons l'hypothèse qu'une CLA d'ordre faible TTI ne dépend pas de δ . Pour faciliter la construction de la CLA TTI, nous allons donc supposer que $\delta = \varepsilon$. Supposer $\delta = \varepsilon$ correspond à ce que l'on appelle l'anisotropie elliptique. Pour être honnête, nous devons préciser qu'il n'existe pas de milieu de ce type mais nous avons pour objectif de construire une CLA d'ordre peu élevé. Notre hypothèse consiste donc à approcher le milieu par un milieu artificiel aux caractéristiques plus simples.

Courbes de lenteur

Comme nous l'avons dit, les propriétés algébriques du système élastique anisotrope sont complexes et nous ne voyons pas comment les exploiter dans le cadre d'un calcul de valeurs propres. Nous nous proposons donc d'étudier plus en détail les propriétés géométriques du système et pour cela, nous pensons que les courbes de lenteur sont un bon sujet. La notion de courbe de lenteur est utilisée pour l'analyse de PML (par exemple dans [BFJ03]). L'idée de les utiliser pour construire des CLA a été suggérée dans [SG10a, SG10b, SG12a, SG12b]. Nous pouvons citer aussi [BGH10] où les courbes de lenteur sont utilisées pour construire des CLA anisotropes pour l'acoustique.

Les courbes de lenteur sont très commodes pour représenter l'anisotropie d'un milieu. Nous appelons vecteur de lenteur le vecteur porté par la direction de propagation de l'onde dont le module est égal à l'inverse de la vitesse de phase. Les courbes de lenteur (surfaces en 3D) sont le lieu des extrémités de ce vecteur de lenteur pour toutes les directions de propagation des ondes. Dans un milieu isotrope par exemple, la vitesse de propagation d'une onde est la même dans toutes les directions. La courbe de lenteur est donc un cercle (une sphère en 3D).

La Fig. 3.1 représente ces formes sur des cas isotrope, VTI et TTI, en différenciant les configurations elliptique et non-elliptique. Elles sont calculées à partir du logiciel MapleTM (voir annexe A.5). Les courbes de lenteur des ondes S forment un cercle dans les cas isotrope et anisotrope elliptique. Par contre, les courbes de lenteur des ondes P forment un cercle dans le cas isotrope, une ellipse dans le cas VTI elliptique et une ellipse inclinée dans le cas TTI elliptique. Toutes ces formes sont paramétrables facilement à partir des données physiques du système de l'élastodynamique : $\rho, V_p, V_s, \varepsilon, \theta$.

La méthode de construction de la CLA TTI s'appuie sur la formulation d'un changement de variables basé sur des géométries simples comme le cercle ou l'ellipse qui décrivent les courbes de lenteur associées au problème de propagation considéré. Nous choisissons comme référence le cas isotrope et les solutions du problème correspondant sont repérées grâce au symbole étoile. Soit $(\mathbf{v}^*, \underline{\sigma}^*)$ la solution du problème de propagation d'ondes élastiques isotropes et $(\mathbf{v}, \underline{\sigma})$ celle du problème anisotrope TTI elliptique. Nous allons exprimer la relation entre ces deux solutions et nous l'utiliserons ensuite pour déduire une CLA TTI d'une CLA isotrope. Les géométries simples des courbes de lenteur fournissent un premier changement

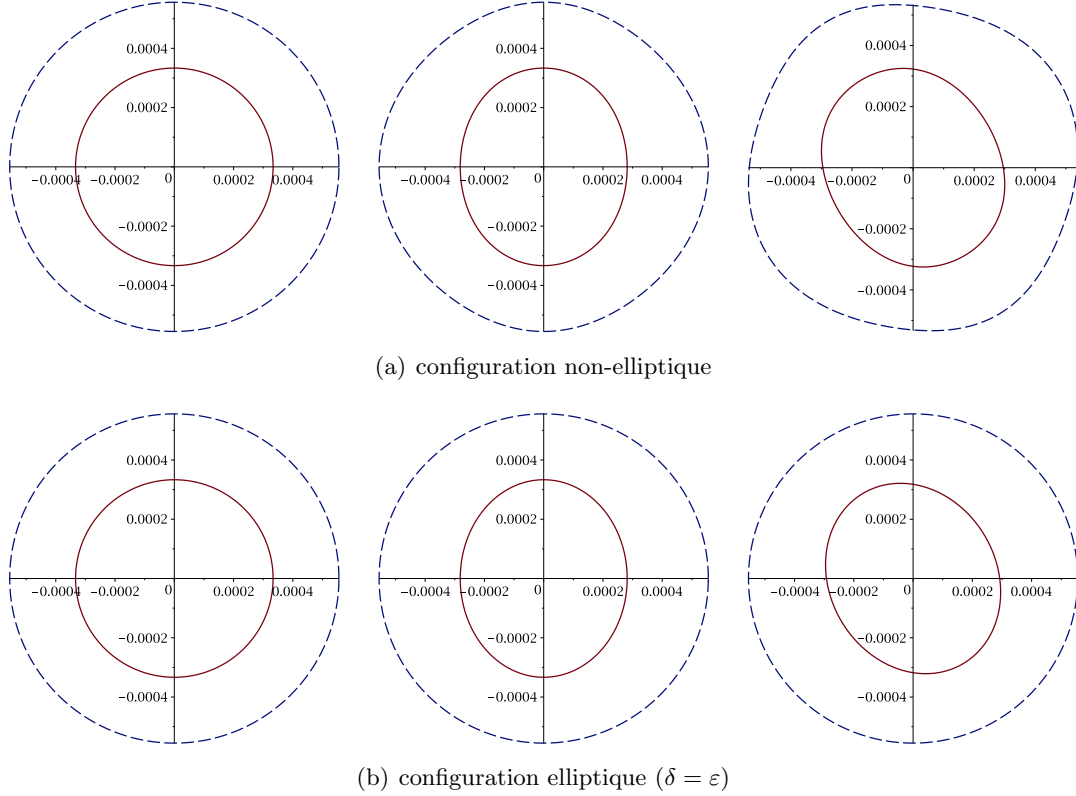


FIGURE 3.1 – Courbes de vitesse élastiques : isotropes (à gauche), VTI (au milieu) et TTI (à droite) avec $\rho = 1 \text{ kg.m}^{-2}$, $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1800 \text{ m.s}^{-1}$, $\varepsilon = 0.20$, $\delta = 0.15$ et $\theta = 30^\circ$, pour les ondes P en rouge trait plein et pour les ondes S en bleu trait pointillé

de variables entre les modes de propagations des systèmes isotrope (k_x^*, k_z^*) et TTI elliptique (k_x, k_z). Il reste à déterminer par identification les changements de variables sur les vitesses \mathbf{v} et \mathbf{v}^* et les tenseurs des contraintes $\underline{\underline{\sigma}}$ et $\underline{\underline{\sigma}}^*$.

CLA TTI elliptique pour les ondes P

Les courbes de vitesse des ondes P et des ondes S sont découplées, ne permettant pas de formuler un changement de variables sur les deux formes géométriques en même temps (voir Fig. 3.1). De plus, les CLA d'ordre faible isotrope (2.24) et VTI (2.53) peuvent être obtenues en découplant les ondes P et les ondes S, comme nous l'avons vu au chapitre précédent à la section 2.2.2. Nous allons donc découpler les problèmes pour construire des CLA TTI pseudo-acoustiques en ondes P et en ondes S.

Nous avons vu que la CLA d'ordre faible VTI pour des ondes S (2.80) est identique à la CLA isotrope puisqu'elle ne fait pas intervenir de coefficients anisotropes. De plus, les courbes de vitesse des ondes S sont les mêmes en isotrope, en VTI elliptique et en TTI elliptique i.e. un cercle de rayon $1/V_s$. Il est donc pertinent de supposer que la CLA d'ordre faible TTI pour les ondes S est inchangée. La suite des développements se concentre donc sur la construction d'une CLA d'ordre faible TTI elliptique pour les ondes P, c'est-à-dire en fixant temporairement $V_s = 0$.

Étape 1 : changement de variables sur les modes de propagation à l'aide des formes géométriques sur les courbes de lenteur. Il s'agit de lier le vecteur d'onde $\mathbf{k} = (k_x, k_z)$ du milieu TTI elliptique au vecteur d'onde $\mathbf{k}^* = (k_x^*, k_z^*)$ du milieu isotrope. Nous proposons le changement de variables suivant :

$$\begin{cases} k_x &= \mu_1 k_x^* + \mu_2 k_z^* \\ k_z &= \mu_3 k_x^* + \mu_4 k_z^* \end{cases} \quad (3.6)$$

où $\{\mu_j\}, 1 \leq j \leq 4$ désignent les coefficients à déterminer.

En rappelant que les composantes k_x et k_z du vecteur d'onde peuvent s'interpréter comme les symboles des dérivées ∂_x et ∂_z , ce changement de variables peut être réinterprété comme un changement de variables dans le plan (x, z) de la forme :

$$\begin{cases} x^* &= \mu_1 x + \mu_3 z \\ z^* &= \mu_2 x + \mu_4 z \end{cases} \quad (3.7)$$

Nous avons choisi de construire une CLA posée sur la frontière $\{x = 0\}$ de normale extérieure \mathbf{e}_x . La frontière artificielle étant une donnée du problème, le changement de variables ne doit pas la modifier et donc préserver la droite d'équation $x = 0$. Appliquer une rotation d'angle TTI θ n'est pas compatible avec cette contrainte. Nous devons donc construire une bijection entre un cercle et une ellipse inclinée conservant la droite $x = 0$. Pour cela, il faut éliminer le coefficient agissant sur les dérivées en z . Nous imposons donc $\mu_3 = 0$. Pour une CLA agissant sur une frontière de normale extérieure \mathbf{e}_z , les dérivées en x seraient à préserver en fixant $\mu_1 = 0$.

En résumé, le changement de variables à déterminer est finalement :

$$\begin{cases} k_x &= \mu_1 k_x^* + \mu_2 k_z^* \\ k_z &= \mu_4 k_z^* \end{cases} \quad (3.8)$$

De plus, nous imposons que le déterminant du système (3.8) soit positif pour conserver l'orientation du repère, c'est-à-dire $\mu_1 \mu_4 > 0$. Au final, nous cherchons la combinaison d'une dilatation d'axe \mathbf{e}_x de rapport μ_1 , d'une dilatation d'axe \mathbf{e}_z de rapport μ_4 et d'une transvection telle que

$$\begin{pmatrix} \mu_1 & \mu_2 \\ 0 & \mu_4 \end{pmatrix} = \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_4 \end{pmatrix} \begin{pmatrix} 1 & \frac{\mu_2}{\mu_1} \\ 0 & 1 \end{pmatrix} \quad (3.9)$$

Les formes géométriques de la Fig. 3.1 se paramétrisent facilement :

- Les modes de propagation des ondes élastiques isotropes (k_x^*, k_z^*) satisfont l'équation quadratique :

$$\rho V_p^2 k_x^{*2} + \rho V_p^2 k_z^{*2} = 1 \quad (3.10)$$

- Les modes de propagation des ondes élastiques TTI elliptique (k_x, k_z) satisfont l'équation générique :

$$\xi_1 k_x^2 + \xi_2 k_x k_z + \xi_3 k_z^2 = 1 \quad (3.11)$$

avec $\kappa = \sqrt{1 + 2\varepsilon}$ et

$$\begin{cases} \xi_1 &= \rho V_p^2 (\kappa^2 \cos^2 \theta + \sin^2 \theta) \\ \xi_2 &= 2 \cos \theta \sin \theta \rho V_p^2 (\kappa^2 - 1) \\ \xi_3 &= \rho V_p^2 (\kappa^2 \sin^2 \theta + \cos^2 \theta). \end{cases} \quad (3.12)$$

En substituant les équations du système (3.8) dans la relation (3.11), il vient :

$$\xi_1 \mu_1^2 k_x^{*2} + \mu_1 (2\xi_1 \mu_2 + \xi_2 \mu_4) k_x^* k_z^* + (\xi_1 \mu_2^2 + \xi_2 \mu_2 \mu_4 + \xi_3 \mu_4^2) k_z^{*2} = 1 \quad (3.13)$$

Par identification de (3.13) avec l'équation du cercle (3.10), le changement de variables sur les modes de propagation (3.8) est déterminé :

$$\begin{cases} \mu_1^2 &= \rho V_p^2 / \xi_1 \\ \mu_2 &= -\xi_2 \mu_4 / 2\xi_1 \\ \mu_4^2 &= \rho V_p^2 / (-\frac{\xi_2^2}{4\xi_1} + \xi_3) \end{cases} \quad (3.14)$$

Nous obtenons finalement :

$$\begin{cases} \mu_1 &= \sqrt{1/(\kappa^2 \cos^2 \theta + \sin^2 \theta)} \\ \mu_2 &= -(\kappa^2 - 1) \cos \theta \sin \theta / \kappa \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \\ \mu_3 &= 0 \\ \mu_4 &= \frac{1}{\kappa} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \end{cases} \quad (3.15)$$

où nous avons fixé arbitrairement $\mu_1 > 0$ et $\mu_4 > 0$, ce qui respecte bien $\mu_1 \mu_4 > 0$.

Étape 2 : changement de variables sur la vitesse. Les notations (v_x^*, v_z^*) et (v_x, v_z) représentent les composantes de la variable vitesse pour les systèmes isotrope et TTI elliptique respectivement. En procédant comme à l'étape 1, nous proposons le changement de variables :

$$\begin{cases} v_x &= \alpha_1 v_x^* + \alpha_2 v_z^* \\ v_z &= \alpha_3 v_x^* + \alpha_4 v_z^* \end{cases} \quad (3.16)$$

où $\{\alpha_j\}, 1 \leq j \leq 4$ désignent les coefficients à déterminer. Ceci se réécrit de façon matricielle comme :

$$\mathbf{v} = A \mathbf{v}^* \quad (3.17)$$

avec $\mathbf{v} = (v_x, v_z)$ et avec la matrice :

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{pmatrix} \quad (3.18)$$

En utilisant le système de l'élastodynamique au second ordre, nous faisons disparaître la variable auxiliaire du tenseur des contraintes, ce qui permet de ne plus considérer que la variable vitesse. Pour un matériau TTI elliptique, ce système devient par transformée de Fourier partielle en x et z :

$$\begin{cases} \rho \partial_t^2 v_x &= C_{11} k_x^2 v_x + C_{13} k_x k_z v_z + C_{15} (k_x^2 v_z + k_z k_x v_x) \\ &+ C_{15} k_x k_z v_x + C_{35} k_z^2 v_z + C_{55} (k_x k_z v_z + k_z^2 v_x) \\ \rho \partial_t^2 v_z &= C_{15} k_x^2 v_x + C_{35} k_x k_z v_z + C_{55} (k_x^2 v_z + k_z k_x v_x) \\ &+ C_{13} k_x k_z v_x + C_{33} k_z^2 v_z + C_{35} (k_x k_z v_z + k_z^2 v_x) \end{cases} \quad (3.19)$$

3.1. Processus de construction en 2D

où C désigne la matrice d'élasticité TTI (3.1) calculée pour le cas elliptique (ie $\delta = \varepsilon$) et en fixant $V_s = 0$.

Ce système s'écrit sous forme matricielle :

$$\rho \partial_t^2 \mathbf{v} = [C_{xx} k_x^2 + C_{xz} k_x k_z + C_{zz} k_z^2] \mathbf{v} \quad (3.20)$$

avec

$$C_{xx} = \begin{pmatrix} C_{11} & C_{15} \\ C_{15} & C_{55} \end{pmatrix}, \quad C_{xz} = \begin{pmatrix} 2C_{15} & (C_{13} + C_{55}) \\ (C_{13} + C_{55}) & 2C_{35} \end{pmatrix}, \quad C_{zz} = \begin{pmatrix} C_{55} & C_{35} \\ C_{35} & C_{33} \end{pmatrix} \quad (3.21)$$

Pour un matériau isotrope et en fixant $V_s = 0$, le système de l'élastodynamique au second ordre se simplifie après transformée de Fourier partielle en x et z en :

$$\begin{cases} \rho \partial_t^2 v_x^* &= \rho V_p^2 (k_x^{*2} v_x^* + k_x^* k_z^* v_z^*) \\ \rho \partial_t^2 v_z^* &= \rho V_p^2 (k_z^* k_x^* v_x^* + k_z^{*2} v_z^*) \end{cases} \quad (3.22)$$

Ce système s'écrit sous la forme matricielle suivante :

$$\rho \partial_t^2 \mathbf{v}^* = \rho V_p^2 [I_{xx} k_x^{*2} + I_{xz} k_x^* k_z^* + I_{zz} k_z^{*2}] \mathbf{v}^* \quad (3.23)$$

avec $\mathbf{v}^* = (v_x^*, v_z^*)$ et

$$I_{xx} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad I_{xz} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad I_{zz} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (3.24)$$

Nous poursuivons ensuite en exprimant (k_x, k_z) en fonction de (k_x^*, k_z^*) grâce à (3.8) et (3.15) puis en remplaçant (v_x, v_z) grâce à (3.16) et en injectant le tout dans le système (3.20) :

$$\rho \partial_t^2 A \mathbf{v}^* = [C_{xx} (\mu_1 k_x^* + \mu_2 k_z^*)^2 + C_{xz} \mu_4 k_z^* (\mu_1 k_x^* + \mu_2 k_z^*) + C_{zz} \mu_4^2 k_z^{*2}] A \mathbf{v}^* \quad (3.25)$$

et qui se réécrit :

$$\rho \partial_t^2 A \mathbf{v}^* = [\mu_1^2 C_{xx} k_x^{*2} + (2\mu_1 \mu_2 C_{xx} + \mu_1 \mu_4 C_{xz}) k_x k_z + (\mu_2^2 C_{xx} + \mu_2 \mu_4 C_{xz} + \mu_4^2 C_{zz}) k_z^{*2}] A \mathbf{v}^* \quad (3.26)$$

Par construction, le système (3.26) doit être équivalent au système isotrope (3.23). Nous procédons donc à l'identification terme à terme qui conduit à un système de trois équations matricielles de la forme :

$$\begin{cases} A^{-1} \mu_1^2 C_{xx} A &= \rho V_p^2 I_{xx} \end{cases} \quad (3.27a)$$

$$\begin{cases} A^{-1} (2\mu_1 \mu_2 C_{xx} + \mu_1 \mu_4 C_{xz}) A &= \rho V_p^2 I_{xz} \end{cases} \quad (3.27b)$$

$$\begin{cases} A^{-1} (\mu_2^2 C_{xx} + \mu_2 \mu_4 C_{xz} + \mu_4^2 C_{zz}) A &= \rho V_p^2 I_{zz} \end{cases} \quad (3.27c)$$

Remarquons que si A est solution de (3.27), alors λA est également solution, pour tout $\lambda \in \mathbb{R}^*$. Le changement de variables est donc déterminé à une constante multiplicative près. Nous introduisons :

$$\begin{cases} a &= -\sqrt{\rho} V_p (\kappa \cos^2 \theta + \sin^2 \theta) / \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \\ b &= \sqrt{\rho} V_p (\kappa - 1) \cos \theta \sin \theta / \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \end{cases} \quad (3.28)$$

et les matrices intervenant dans les équations (3.27a) et (3.27c) peuvent se réécrire :

$$\begin{cases} A^{-1} \begin{pmatrix} a^2 & ab \\ ab & b^2 \end{pmatrix} A &= \rho V_p^2 I_{xx} \\ A^{-1} \begin{pmatrix} b^2 & -ab \\ -ab & a^2 \end{pmatrix} A &= \rho V_p^2 I_{zz} \end{cases} \quad (3.29)$$

ce qui conduit aux relations :

$$\begin{cases} \alpha_2 &= -\alpha_4 b/a \\ \alpha_3 &= \alpha_1 b/a \end{cases} \quad (3.30)$$

En se servant de ces relations, l'équation (3.27b) donne :

$$\alpha_1 = \alpha_4 \quad (3.31)$$

Il reste une indéterminée qui traduit le fait que le changement est déterminé à une constante près. Si nous fixons arbitrairement $\alpha_1 = a$, la matrice de passage devient :

$$A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \quad (3.32)$$

Le changement de variables sur les vitesses (3.16) s'écrit alors :

$$\begin{cases} v_x &= \frac{\sqrt{\rho} V_p}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [-(\kappa \cos^2 \theta + \sin^2 \theta) v_x^* - (\kappa - 1) \cos \theta \sin \theta v_z^*] \\ v_z &= \frac{\sqrt{\rho} V_p}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [(\kappa - 1) \cos \theta \sin \theta v_x^* - (\kappa \cos^2 \theta + \sin^2 \theta) v_z^*] \end{cases} \quad (3.33)$$

Étape 3 : changement de variables sur le tenseur des contraintes. De la même manière que précédemment, $\underline{\sigma}^*$ et $\underline{\sigma}$ représentent les tenseurs des contraintes des systèmes isotrope et TTI elliptique respectivement. Nous posons le changement de variables :

$$\begin{cases} \sigma_{xx} &= \beta_1 \sigma_{xx}^* + \beta_2 \sigma_{zz}^* + \beta_3 \sigma_{xz}^* \\ \sigma_{zz} &= \beta_4 \sigma_{xx}^* + \beta_5 \sigma_{zz}^* + \beta_6 \sigma_{xz}^* \\ \sigma_{xz} &= \beta_7 \sigma_{xx}^* + \beta_8 \sigma_{zz}^* + \beta_9 \sigma_{xz}^* \end{cases} \quad (3.34)$$

où $\{\beta_j\}_{1 \leq j \leq 9}$ désignent les coefficients à déterminer.

Dans le système de l'élastodynamique isotrope pour les ondes P seules (i.e. en fixant $V_s = 0$), le tenseur des contraintes se résume à une seule variable p^* , ce qui simplifie le problème (3.34) car $\sigma_{xx}^* = \sigma_{zz}^* = p^*$ et $\sigma_{xz}^* = 0$. Il ne reste donc plus qu'à déterminer trois coefficients, notés $\{\tilde{\beta}_j\}_{1 \leq j \leq 3}$, tels que :

$$\begin{cases} \sigma_{xx} &= \tilde{\beta}_1 p^* \\ \sigma_{zz} &= \tilde{\beta}_2 p^* \\ \sigma_{xz} &= \tilde{\beta}_3 p^* \end{cases} \quad (3.35)$$

3.1. Processus de construction en 2D

En procédant comme dans l'étape 2, il suffit d'écrire le système de l'élastodynamique (1.5) pour un matériau isotrope en fixant $V_s = 0$ et en appliquant une transformée de Fourier partielle en x et z :

$$\begin{cases} \rho \partial_t v_x^* &= k_x^* p^* \\ \rho \partial_t v_z^* &= k_z^* p^* \\ \partial_t p^* &= \rho V_p^2 k_x^* v_x^* + \rho V_p^2 k_z^* v_z^* \end{cases} \quad (3.36)$$

puis pour un matériau TTI :

$$\begin{cases} \rho \partial_t v_x &= k_x \sigma_{xx} + k_z \sigma_{xz} \\ \rho \partial_t v_z &= k_x \sigma_{xz} + k_z \sigma_{zz} \\ \partial_t \sigma_{xx} &= C_{11} k_x v_x + C_{13} k_z v_z + C_{15} (k_x v_z + k_z v_x) \\ \partial_t \sigma_{zz} &= C_{13} k_x v_x + C_{33} k_z v_z + C_{35} (k_x v_z + k_z v_x) \\ \partial_t \sigma_{xz} &= C_{15} k_x v_x + C_{35} k_z v_z + C_{55} (k_x v_z + k_z v_x) \end{cases} \quad (3.37)$$

où C est la matrice d'élasticité TTI (3.1) calculée en elliptique (ie $\delta = \varepsilon$) en fixant $V_s = 0$.

Puis, en substituant les changements de variables : sur les modes de propagation (3.8) obtenu à l'étape 1, sur les vitesses (3.33) obtenu à l'étape 2 et sur les contraintes (3.35) dans le système TTI elliptique (3.37), nous obtenons un système en $\sigma_{xx}^*, \sigma_{zz}^*, \sigma_{xz}^*$ de la forme :

$$\begin{cases} \rho \partial_t (\alpha_1 v_x^* + \alpha_2 v_z^*) &= (\mu_1 k_x^* + \mu_2 k_z^*) \tilde{\beta}_1 p^* + \mu_4 k_z^* \tilde{\beta}_3 p^* \\ \rho \partial_t (\alpha_3 v_x^* + \alpha_4 v_z^*) &= (\mu_1 k_x^* + \mu_2 k_z^*) \tilde{\beta}_3 p^* + \mu_4 k_z^* \tilde{\beta}_2 p^* \\ \partial_t \tilde{\beta}_1 p^* &= C_{11} (\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_1 v_x^* + \alpha_2 v_z^*) + C_{13} \mu_4 k_z^* (\alpha_3 v_x^* + \alpha_4 v_z^*) \\ &\quad + C_{15} ((\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_3 v_x^* + \alpha_4 v_z^*) + \mu_4 k_z^* (\alpha_1 v_x^* + \alpha_2 v_z^*)) \\ \partial_t \tilde{\beta}_2 p^* &= C_{13} (\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_1 v_x^* + \alpha_2 v_z^*) + C_{33} \mu_4 k_z^* (\alpha_3 v_x^* + \alpha_4 v_z^*) \\ &\quad + C_{35} ((\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_3 v_x^* + \alpha_4 v_z^*) + \mu_4 k_z^* (\alpha_1 v_x^* + \alpha_2 v_z^*)) \\ \partial_t \tilde{\beta}_3 p^* &= C_{15} (\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_1 v_x^* + \alpha_2 v_z^*) + C_{35} \mu_4 k_z^* (\alpha_3 v_x^* + \alpha_4 v_z^*) \\ &\quad + C_{55} ((\mu_1 k_x^* + \mu_2 k_z^*) (\alpha_3 v_x^* + \alpha_4 v_z^*) + \mu_4 k_z^* (\alpha_1 v_x^* + \alpha_2 v_z^*)) \end{cases} \quad (3.38)$$

qui doit être équivalent au système isotrope (3.36).

En procédant par identification dans les trois dernières équations, nous pouvons déterminer le changement de variables sur les tenseurs des contraintes (3.35) :

$$\begin{cases} \sigma_{xx} &= -\sqrt{\rho} V_p (\kappa \cos^2 \theta + \sin^2 \theta) p^* \\ \sigma_{zz} &= -\sqrt{\rho} V_p (\kappa \sin^2 \theta + \cos^2 \theta) p^* \\ \sigma_{xz} &= \sqrt{\rho} V_p (\kappa - 1) \cos \theta \sin \theta p^* \end{cases} \quad (3.39)$$

Étape 4 : formulation de la CLA TTI elliptique pour les ondes P. Il suffit d'utiliser la CLA isotrope pour les ondes P déduite de celle du VTI (2.66), soit : $p^* = -\rho V_p v_x^*$. La CLA TTI va s'écrire à partir de la CLA isotrope de la façon suivante. Le terme p^* est remplacé dans (3.39) pour exprimer σ_{xx} et σ_{xz} , qui sont les deux termes résultant du produit $\underline{\underline{\sigma}} \mathbf{n}$ lorsque $\mathbf{n} = \mathbf{e}_x$,

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

en fonction de v_x^* . Puis, le terme v_x^* est exprimé en fonction de v_x et v_z grâce à (3.33). Ce processus conduit à la CLA TTI pour les ondes P :

$$\begin{cases} \sigma_{xx} &= -\rho V_p \frac{\kappa \cos^2 \theta + \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [(\kappa \cos^2 \theta + \sin^2 \theta) v_x - (\kappa - 1) \cos \theta \sin \theta v_z] \\ \sigma_{xz} &= -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [-(\kappa \cos^2 \theta + \sin^2 \theta) v_x + (\kappa - 1) \cos \theta \sin \theta v_z] \end{cases} \quad (3.40)$$

En résumé :

En se plaçant dans le domaine de Fourier en espace, nous exprimons le champ anisotrope (TTI elliptique) en fonction du champ isotrope. Pour cela, **(a)** nous initialisons le procédé en exprimant les vecteurs d'ondes isotropes. L'initialisation peut se faire car nous connaissons l'expression des courbes de lenteur et donc les vecteurs d'ondes ; **(b)** une fois les vecteurs d'ondes connus, nous pouvons exprimer le champ de vitesse après transformée de Fourier en espace – cette étape est facilitée par le fait qu'il est possible d'éliminer $\underline{\sigma}$ grâce à la formulation du second-ordre de l'équation des ondes élastiques ; **(c)** nous connaissons le vecteur d'onde anisotrope et le champ de vitesse anisotrope, nous pouvons en déduire le tenseur $\underline{\underline{\sigma}}$; **(d)** les champs anisotropes s'exprimant en fonction des champs isotropes, nous appliquons la CLA isotrope et nous revenons aux champs anisotropes grâce au changement d'inconnues inverses, ce qui nous permet finalement d'obtenir une CLA anisotrope (TTI elliptique).

CLA pour le système de l'élastodynamique TTI elliptique

Pour former une CLA TTI pour le système de l'élastodynamique, c'est-à-dire sans découpler les ondes P et les ondes S, nous proposons d'additionner les contributions des CLA pour les ondes P elliptiques (3.40) et pour les ondes S (2.80). Nous avons vu à la section 2.2.2 que la CLA d'ordre faible VTI est égale à la somme des contributions des CLA des sous-problèmes pseudo-acoustiques en ondes P et ondes S et c'est ce qui nous incite à proposer cette approche. En effet, nous savons qu'un milieu TTI peut être vu comme une transformation simple (à savoir une rotation) d'un milieu VTI. Nous pouvons donc supposer que la CLA complète pour le cas TTI s'obtient comme la CLA complète pour le cas VTI. Cette somme donne :

$$\begin{cases} \sigma_{xx} &= -\rho V_p \frac{\kappa \cos^2 \theta + \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [(\kappa \cos^2 \theta + \sin^2 \theta) v_x - (\kappa - 1) \cos \theta \sin \theta v_z] \\ \sigma_{xz} &= -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [-(\kappa \cos^2 \theta + \sin^2 \theta) v_x + (\kappa - 1) \cos \theta \sin \theta v_z] - \rho V_s v_z \end{cases} \quad (3.41)$$

Remarque 3.1 Lorsque l'angle d'inclinaison est nul (i.e. $\theta = 0$), la CLA TTI (3.41) coïncide avec la CLA VTI (2.53).

Remarque 3.2 Lorsque les constantes de Thomsen sont nulles (i.e. $\varepsilon = \delta = 0$, soit $\kappa = 1$), la CLA TTI (3.41) redevient isotrope, c'est-à-dire comme (2.24), quel que soit l'angle d'inclinaison. Ceci est dû au fait qu'en isotrope, les vitesses de propagation sont les mêmes dans toutes les directions.

3.1.3 Étude expérimentale pour l'élastique TTI 2D

Cas tests

Pour commencer, nous reprenons les cas tests simples déjà décrits dans les cas isotrope et VTI, à savoir un cas homogène, c'est-à-dire avec un seul matériau, puis des cas bi-couches, de type bande ou coin, voir Fig. 3.2, de dimension 10 km^2 dont les caractéristiques physiques sont décrites dans le tableau 3.1 (où nous considérons que la masse volumique en 2D est une masse surfacique).

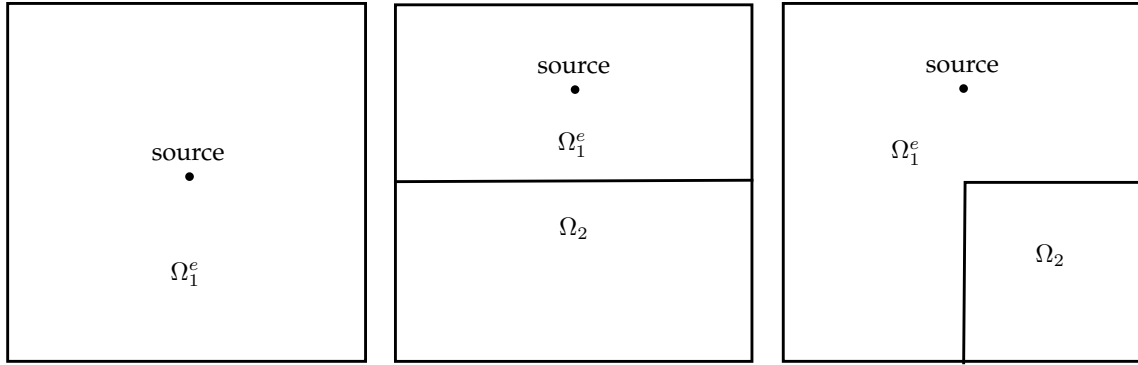


FIGURE 3.2 – Cas tests TTI 2D homogène (gauche), bi-couches (centre) et coin (droite)

milieu	ρ [kg.m ⁻²]	V_p [m.s ⁻¹]	V_s [m.s ⁻¹]	ϵ	δ	θ [°]
Ω_1	1	3000	1800	0.20	0.10	30
Ω_1^e	1	3000	1800	0.20	0.20	30
Ω_2	1	2200	1250	0.15	0.05	15

TABLE 3.1 – Caractéristiques physiques des cas-tests TTI 2D

Source et condition initiale

De même que pour les cas isotrope et VTI, nous allons suivre la procédure décrite dans la section 1.1.2 pour construire une condition initiale ou une source afin de ne générer qu'une onde P, qu'une onde S ou la combinaison d'une onde P et d'une onde S.

Dans un milieu TTI 2D, la matrice de l'équation de dispersion s'écrit :

$$E_{TTI,2D} = \begin{pmatrix} C_{11}k_x^2 + 2C_{15}k_xk_z + C_{55}k_z^2 & C_{15}k_x^2 + (C_{55} + C_{13})k_xk_z + C_{35}k_z^2 \\ C_{15}k_x^2 + (C_{55} + C_{13})k_xk_z + C_{35}k_z^2 & C_{55}k_x^2 + 2C_{35}k_xk_z + C_{33}k_z^2 \end{pmatrix} \quad (3.42)$$

où C est la matrice d'élasticité TTI 2D (3.1).

Dans le cas elliptique, les vecteurs propres de (3.42) sont orthogonaux et font intervenir

les symboles de Fourier :

$$\begin{cases} \Lambda_p &= \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta) i k_x + (X - Z) \cos \theta \sin \theta i k_z \\ (X - Z) \cos \theta \sin \theta i k_x + (Z \cos^2 \theta + X \sin^2 \theta) i k_z \end{pmatrix} \\ \Lambda_s &= \begin{pmatrix} -(X - Z) \cos \theta \sin \theta i k_x - (Z \cos^2 \theta + X \sin^2 \theta) i k_z \\ (X \cos^2 \theta + Z \sin^2 \theta) i k_x + (X - Z) \cos \theta \sin \theta i k_z \end{pmatrix} \end{cases} \quad (3.43)$$

ce qui conduit aux opérateurs :

$$\begin{cases} B_p &= \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta) \partial_x + (X - Z) \cos \theta \sin \theta \partial_z \\ (X - Z) \cos \theta \sin \theta \partial_x + (Z \cos^2 \theta + X \sin^2 \theta) \partial_z \end{pmatrix} \\ B_s &= \begin{pmatrix} -(X - Z) \cos \theta \sin \theta \partial_x - (Z \cos^2 \theta + X \sin^2 \theta) \partial_z \\ (X \cos^2 \theta + Z \sin^2 \theta) \partial_x + (X - Z) \cos \theta \sin \theta \partial_z \end{pmatrix} \end{cases} \quad (3.44)$$

avec $X = \sqrt{\kappa^2 V_p^2 - V_s^2}$ et $Z = \sqrt{V_p^2 - V_s^2}$.

Nous rappelons que nous avons choisi pour conditions initiales $\underline{\sigma}_0 = 0$ et $\mathbf{v}_0(\mathbf{x}) = B_\star(e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2})$. Pour l'équation élastique TTI elliptique 2D, cela conduit aux différents choix :

$$\begin{cases} \mathbf{v}_0(\mathbf{x}) &= \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta)(x - x_0) + (X - Z) \cos \theta \sin \theta (z - z_0) \\ (X - Z) \cos \theta \sin \theta (x - x_0) + (Z \cos^2 \theta + X \sin^2 \theta)(z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \\ &\text{pour ne générer qu'une onde P,} \\ \mathbf{v}_0(\mathbf{x}) &= \begin{pmatrix} -(X - Z) \cos \theta \sin \theta (x - x_0) - (Z \cos^2 \theta + X \sin^2 \theta)(z - z_0) \\ (X \cos^2 \theta + Z \sin^2 \theta)(x - x_0) + (X - Z) \cos \theta \sin \theta (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \\ &\text{pour ne générer qu'une onde S,} \end{cases} \quad (3.45)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Pour utiliser une source ponctuelle à la place d'une condition initiale, nous rappelons que sa forme, lorsqu'elle est appliquée sur l'équation du tenseur des contraintes, est $s_R(t)\underline{s}(\mathbf{x})$, où $s_R(t)$ est une amplitude (en l'occurrence une ondelette de Ricker définie en (1.9)). Il suffit ensuite de chercher \underline{s} telle que $\nabla \cdot \underline{s} = B_\star(\delta_{\mathbf{x}_0})$. Ecrivons cette égalité pour B_p :

$$\begin{pmatrix} \partial_x s_{xx} + \partial_z s_{xz} \\ \partial_x s_{zx} + \partial_z s_{zz} \end{pmatrix} = \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta) \partial_x + (X - Z) \cos \theta \sin \theta \partial_z \\ (X - Z) \cos \theta \sin \theta \partial_x + (Z \cos^2 \theta + X \sin^2 \theta) \partial_z \end{pmatrix} \delta_{\mathbf{x}_0} \quad (3.46)$$

L'identification terme à terme conduit à une solution simple : $s_{xz} = s_{zx} = (X - Z) \cos \theta \sin \theta \delta_{\mathbf{x}_0}$, $s_{xx} = (X \cos^2 \theta + Z \sin^2 \theta) \delta_{\mathbf{x}_0}$ et $s_{zz} = (Z \cos^2 \theta + X \sin^2 \theta) \delta_{\mathbf{x}_0}$. En procédant de la même façon pour B_s , les différentes sources possibles pour l'équation élastique TTI elliptique 2D s'écrivent

finalement :

$$\begin{cases} \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta) & (X - Z) \cos \theta \sin \theta \\ (X - Z) \cos \theta \sin \theta & (Z \cos^2 \theta + X \sin^2 \theta) \end{pmatrix}, & \text{pour ne générer qu'une onde P,} \\ \underline{s}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) \begin{pmatrix} -(X - Z) \cos \theta \sin \theta & -(Z \cos^2 \theta + X \sin^2 \theta) \\ (X \cos^2 \theta + Z \sin^2 \theta) & (X - Z) \cos \theta \sin \theta \end{pmatrix}, & \text{pour ne générer qu'une onde S,} \end{cases} \quad (3.47)$$

ou la somme des deux pour générer une combinaison d'ondes P et S.

Remarque 3.3 Les matrices qui apparaissent dans les sources TTI (3.47) peuvent également se déduire de l'étude VTI. Nous notons les matrices des sources VTI (2.86)

$$M_{VTI,P} = \begin{pmatrix} X & 0 \\ 0 & Z \end{pmatrix}, \quad M_{VTI,S} = \begin{pmatrix} 0 & -Z \\ X & 0 \end{pmatrix}$$

alors les matrices des sources TTI s'obtiennent par la formule

$$M_{TTI,\star} = R_\theta^T M_{VTI,\star} R_\theta, \quad \text{avec } R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

où R_θ est la matrice du changement de base entre le repère cartésien et le repère TTI 2D.

Les opérateurs intervenant dans les conditions initiales TTI (3.45) et donc les vecteurs propres de la matrice de dispersion TTI (3.55) peuvent également se déduire de la même façon :

$$\Lambda_{TTI,\star} = R_\theta^T M_{VTI,\star} R_\theta \begin{pmatrix} ik_x \\ ik_z \end{pmatrix}$$

Résultats numériques

Comme pour les cas isotrope et VTI, dans les trois configurations de domaine décrites à la Fig. 3.2 et dans le tableau 3.1, nous allons considérer une source élastique (ondes P et S) puis des sources pseudo-acoustiques (onde P ou onde S). Les résultats sur le domaine homogène sont présentés Fig. 3.3, ceux des domaines bi-couches bande et coin sont présentés Fig. 3.4 et 3.5 respectivement. Le module du vecteur vitesse est représenté pour différents temps de simulation. À $t = 1s$, nous observons le front d'onde, ce qui permet de figer l'échelle des représentations. À $t = 2s$ ou $t = 3s$, nous commençons à observer le fonctionnement des CLA car le front d'onde a atteint les bords du domaine de calcul. Nous laissons ensuite les calculs continuer jusqu'à $t = 5s$ puis $t = 7s$. Nous observons ainsi les réflexions parasites propres aux bords absorbants. Les mêmes résultats sont ensuite présentés à l'échelle de couleurs des réflexions, calculée au temps, en fonction des cas, $t = 5s$ (homogène) ou $t = 7s$ (bi-couches). Toutes les valeurs des échelles sont regroupées dans le tableau 3.2.

Remarque 3.4 À notre connaissance, il n'existe aucun travail dédié à la construction de source ou conditions initiales pertinentes pour tester les performances des conditions aux limites. C'est étonnant car c'est un point délicat dont nous entendons souvent parler dans l'industrie. Nous ne pouvons donc valider la nouvelle condition aux limites en considérant des résultats de la littérature et c'est pourquoi nous allons plutôt utiliser nos expériences dans des milieux VTI et isotropes.

source	temps	homogène	bi-couche bande	bi-couche coin
PS	1s / 1s / 1s	2.00	2.00	2.00
	5s / 7s / 7s	0.12	0.27	0.27
P	1s / 1s / 1s	1.68	1.85	1.89
	5s / 7s / 7s	0.05	0.12	0.10
S	1s / 1s / 1s	2.00	2.00	2.00
	5s / 7s / 7s	0.12	0.26	0.24

TABLE 3.2 – Valeurs maximales du module du vecteur vitesse pour l’étude TTI 2D

Comparaison avec la CLA isotrope

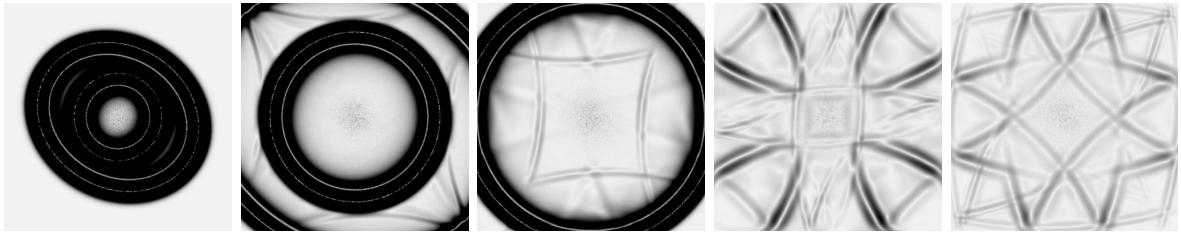
Nous proposons maintenant de comparer la CLA TTI à la CLA isotrope dans le domaine homogène TTI elliptique. Les résultats avec des sources élastiques (ondes P et S) et pseudo-acoustiques (onde P et onde S) sont représentés pour la CLA isotrope Fig. 3.6 (à comparer avec la Fig. 3.3). Il apparaît avant toute chose que les réflexions sont assez faibles même en utilisant une CLA isotrope. La différence la plus importante réside dans la réflexion de l’onde P en onde P qui n’apparaît pas dans le cas de la CLA TTI et qui est très visible dans le cas de la CLA isotrope. Ces réflexions sont plus fortes pour un bord de normale extérieure \mathbf{e}_x que pour un bord de normale extérieure \mathbf{e}_z . Cela vient du fait que l’angle TTI est assez faible et donc la propagation des ondes dans le sens verticale est proche d’un comportement isotrope. La réflexion d’onde P en onde S est aussi plus forte avec une CLA isotrope, ceci peut notamment se vérifier en comparant les échelles de représentation des calculs. Enfin, les réflexions de l’onde S sont comparables et cela est cohérent puisque l’onde S est à chaque fois modélisée de la même façon.

Non-elliptique

Pour construire la CLA TTI, nous avons supposé que le milieu est elliptique (et donc artificiel). Cela a eu pour conséquence de simplifier les calculs en éliminant le paramètre δ et la condition obtenue ne dépend donc pas de ce paramètre. Nous émettons le postulat que la condition obtenue est aussi valable dans un milieu non elliptique (et donc réel) en nous appuyant sur la remarque que la CLA d’ordre faible pour un milieu VTI ne dépend pas non plus du paramètre δ et que les deux milieux s’expriment simplement l’un par rapport à l’autre *via* une rotation. En reprenant l’exemple de configuration non-elliptique dans un domaine homogène avec les paramètres précisés dans le tableau 3.1 et pour une source élastique, nous obtenons les résultats de la Fig. 3.7, qui sont en effet similaires à ceux du cas VTI (Fig. 2.10). La comparaison avec l’utilisation d’une CLA isotrope est présentée Fig. 3.8. De même que pour le cas elliptique, les réflexions de l’onde P sont beaucoup plus fortes avec une CLA isotrope.



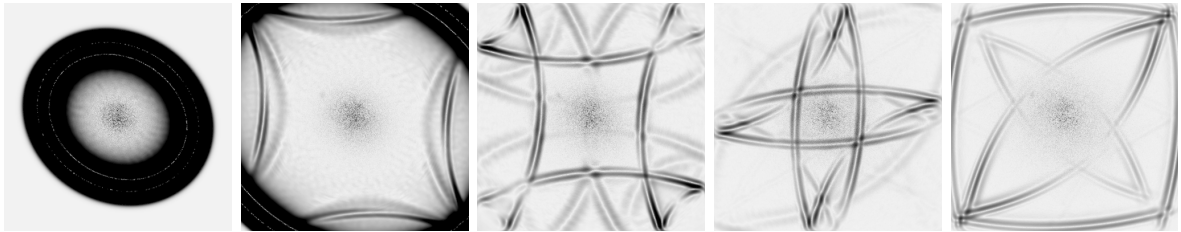
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$



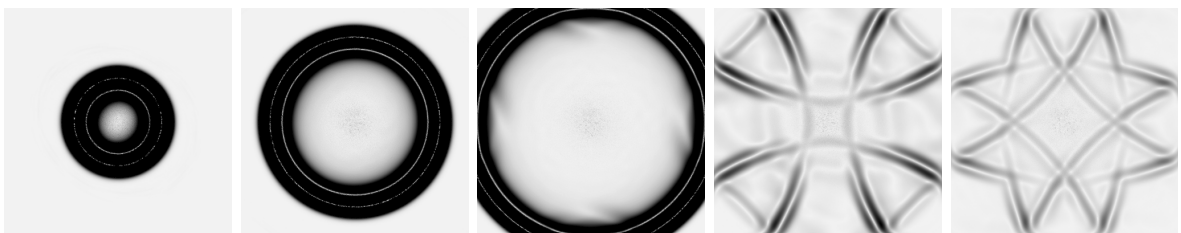
(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 5s$



(e) source S, échelle de couleur fixée à $t = 1s$



(f) source S, échelle de couleur fixée à $t = 5s$

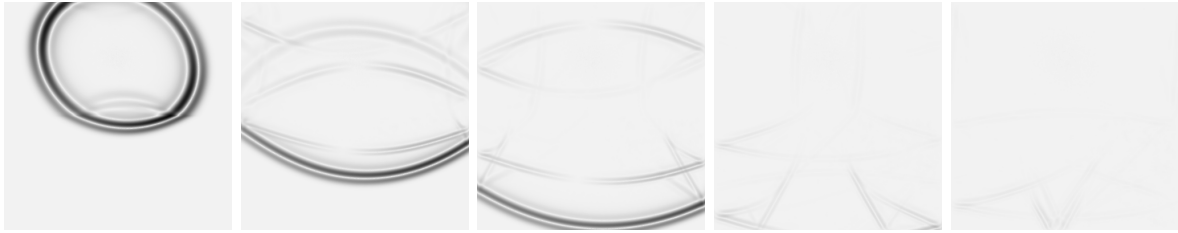
FIGURE 3.3 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration TTI homogène 2D



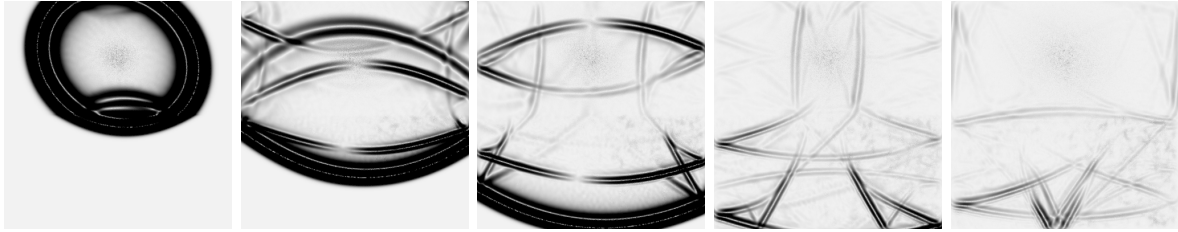
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



(c) source P, échelle de couleur fixée à $t = 1s$



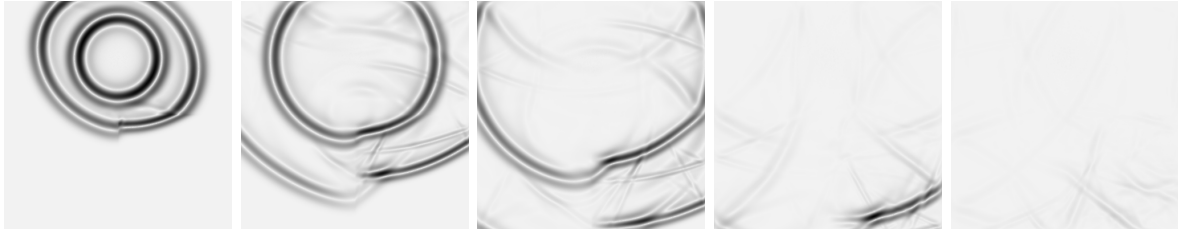
(d) source P, échelle de couleur fixée à $t = 7s$



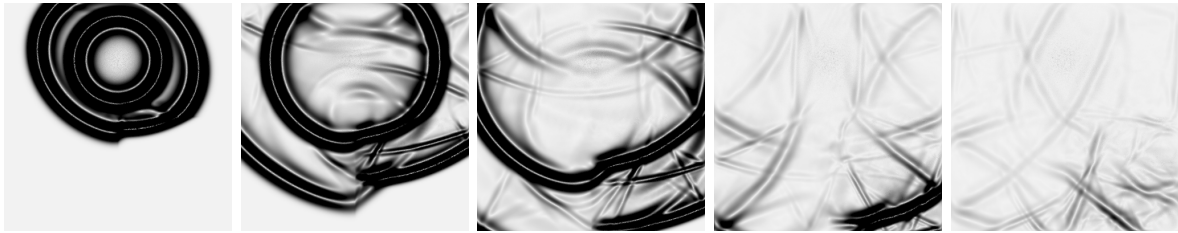
(e) source S, échelle de couleur fixée à $t = 1s$



(f) source S, échelle de couleur fixée à $t = 7s$



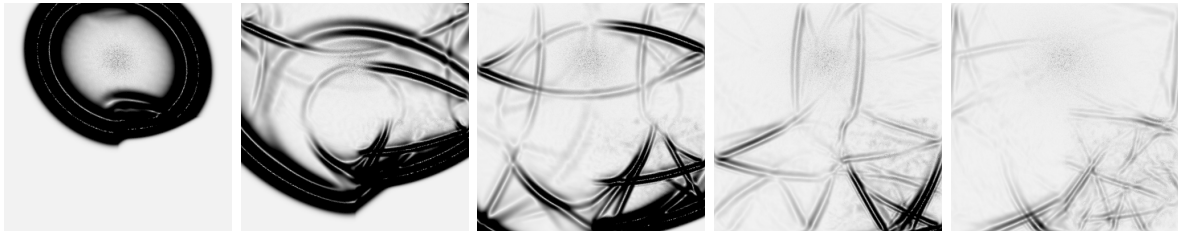
(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 7s$



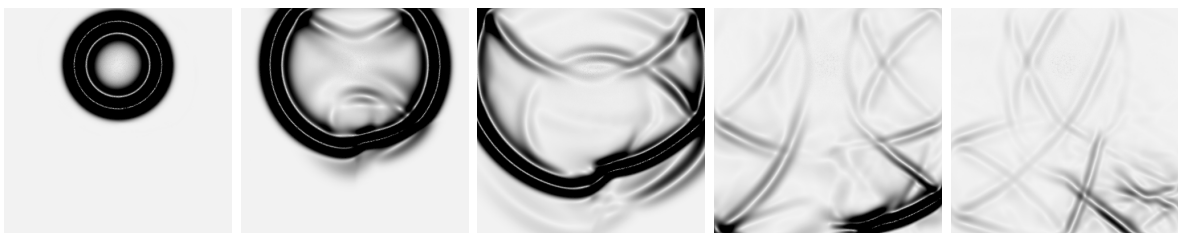
(c) source P, échelle de couleur fixée à $t = 1s$



(d) source P, échelle de couleur fixée à $t = 7s$



(e) source S, échelle de couleur fixée à $t = 1s$

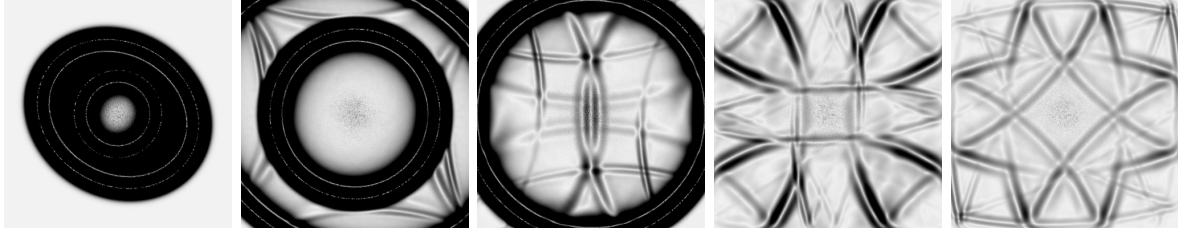


(f) source S, échelle de couleur fixée à $t = 7s$

FIGURE 3.5 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 3s, 5s$, et $7s$ (de gauche à droite) sur la configuration bi-couche coin TTI 2D



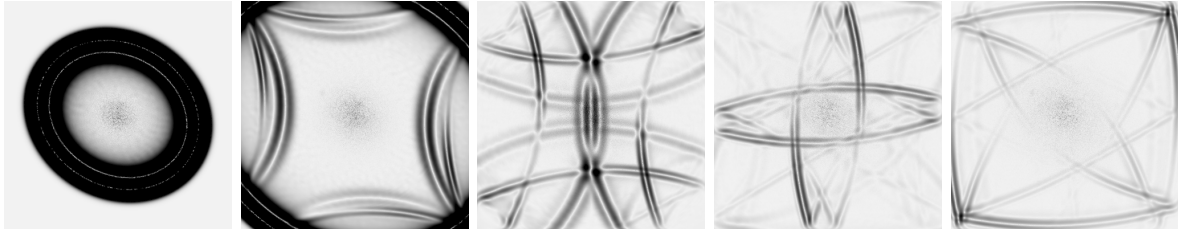
(a) source PS, échelle de couleur fixée à $t = 1s : 2.00$



(b) source PS, échelle de couleur fixée à $t = 5s : 0.14$



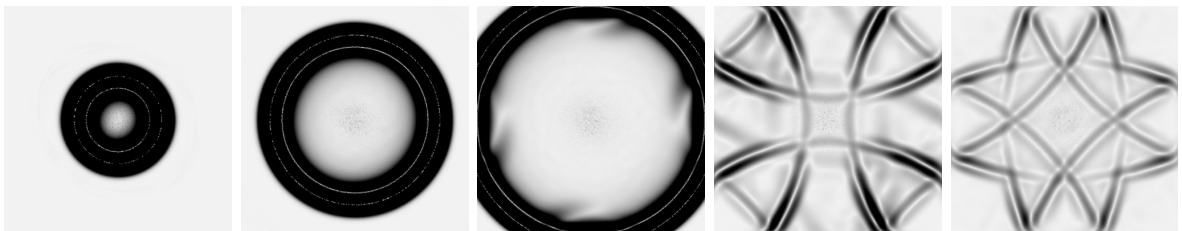
(c) source P, échelle de couleur fixée à $t = 1s : 1.68$



(d) source P, échelle de couleur fixée à $t = 5s : 0.06$



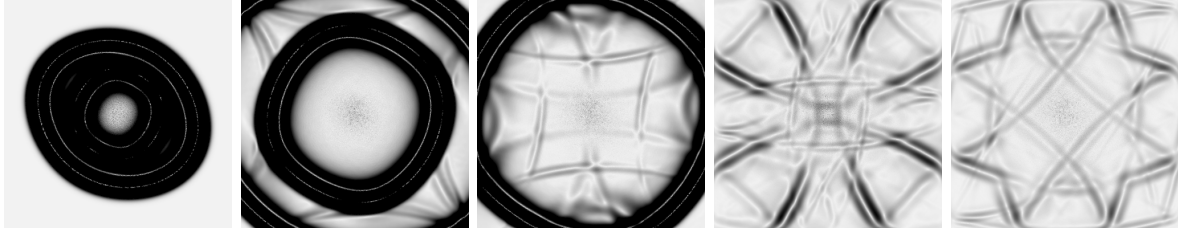
(e) source S, échelle de couleur fixée à $t = 1s : 2.00$



(f) source S, échelle de couleur fixée à $t = 5s : 0.14$



(a) source PS, échelle de couleur fixée à $t = 1s$

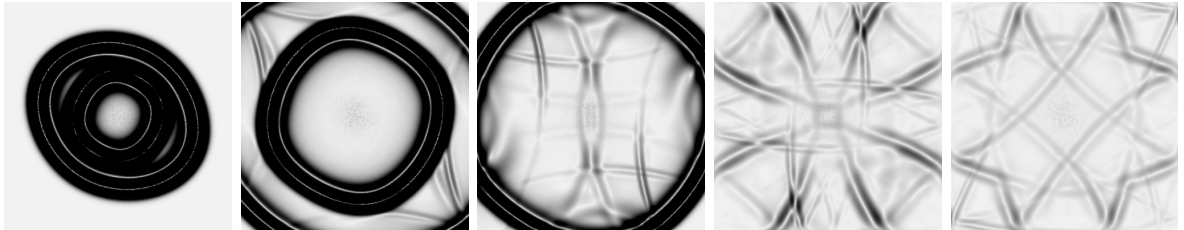


(b) source PS, échelle de couleur fixée à $t = 5s$

FIGURE 3.7 – Module du vecteur vitesse au cours du temps à $t = 1s, 2s, 5s$ et $7s$ pour un cas 2D TTI non-elliptique homogène. Les valeurs maximales sont 2.04 et 0.10 aux temps respectifs $t = 1s$ et $t = 5s$.



(a) source PS, échelle de couleur fixée à $t = 1s$



(b) source PS, échelle de couleur fixée à $t = 5s$

FIGURE 3.8 – Comparaison avec la CLA isotrope à $t = 1s, 2s, 5s$ et $7s$ dans un milieu 2D TTI non-elliptique homogène. Les valeurs maximales sont 2.04 et 0.16 aux temps respectifs $t = 1s$ et $t = 5s$.

3.1.4 Stabilité des problèmes mixtes TTI 2D

Comme à la section 2.3.1, le système de l'élastodynamique (1.5) est lié à la fonction $\mathbb{E}(t) = \frac{1}{2}(\|v\|_{\Omega_x, \rho}^2 + \|\underline{\underline{\sigma}}\|_{\Omega_x, \underline{\underline{C}}^{-1}}^2)$. Nous supposons que la solution du problème est assez régulière pour avoir :

$$\frac{d}{dt}\mathbb{E}(t) = \int_{\Gamma_x} (\underline{\underline{\sigma}}\mathbf{n})v d\mathbf{x} \quad (3.48)$$

Nous nous plaçons sur une frontière $\Gamma_x = \{x = 0\}$. Le vecteur \mathbf{n} est la normale extérieure à Γ_x donc $\mathbf{n} = (1, 0)$ et $\underline{\underline{\sigma}}\mathbf{n} = (\sigma_{xx}, \sigma_{xz})$. Enfin, les composantes de $\underline{\underline{\sigma}}$ sont définies par la CLA TTI du système (3.41). Ainsi :

$$\left\{ \begin{array}{l} \frac{d}{dt}\mathbb{E}(t) = - \int_{\Gamma_x} [\sqrt{\rho V_p} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} [(\kappa \cos^2 \theta + \sin^2 \theta)v_x - (\kappa - 1) \cos \theta \sin \theta v_z]^2 \\ + \rho V_s v_z^2] d\mathbf{x} \end{array} \right. \quad (3.49)$$

Comme cette quantité est négative, l'énergie du système continu est décroissante, ce qui assure la stabilité du problème de l'élastodynamique 2D TTI avec la CLA (3.41).

En suivant les mêmes étapes qu'au chapitre précédent, nous pouvons affirmer que l'énergie

$$\mathbb{E}_{semi_d}(t) = \langle \partial_t M_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h \rangle + \langle \partial_t M_{\boldsymbol{\sigma}, C^{-1}} \boldsymbol{\sigma}_h, \boldsymbol{\sigma}_h \rangle \quad (3.50)$$

associée au problème mixte semi-discret obtenu en appliquant une formulation DG avec flux centrés est décroissante. Enfin, pour que la quantité

$$\mathbb{E}_d^n = \frac{1}{2} \langle M_{\mathbf{v}} \mathbf{v}_h^n, \mathbf{v}_h^n \rangle + \frac{1}{2} \langle M_{\boldsymbol{\sigma}, C^{-1}} \boldsymbol{\sigma}_h^{n+\frac{1}{2}}, \boldsymbol{\sigma}_h^{n-\frac{1}{2}} \rangle \quad (3.51)$$

associée au problème mixte discret obtenu en combinant au problème semi-discret un schéma en temps de type Leap-Frog soit une énergie, il faut et il suffit que la matrice

$$M_{\mathbf{v}} - \frac{\Delta t^2}{4} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C})^{-1} (R_{\boldsymbol{\sigma}})^T \quad (3.52)$$

soit positive, ce qui définit une condition de stabilité du schéma de type CFL

$$1 - \frac{\Delta t^2}{4} \lambda_{max} \geq 0 \quad (3.53)$$

où λ_{max} est la plus grande valeur propre de la matrice $M_{\mathbf{v}}^{-1} R_{\boldsymbol{\sigma}} (M_{\boldsymbol{\sigma}, C^{-1}})^{-1} (R_{\boldsymbol{\sigma}})^T$.

Remarque 3.5 Dans le code DIVA, nous mesurons l'énergie discrète suivant la formule (3.51). Nous avons également modifié la CFL en implémentant la méthode de la puissance (voir algorithme 2.1), qui a l'avantage de s'appliquer indépendamment à des configurations isotropes et anisotropes.

3.1.5 Courbes d'énergies de l'étude TTI 2D

De même que pour les cas isotrope et VTI, la Fig. 3.9 décrit le comportement de l'énergie discrète pour des sources données par une onde P, une onde S et une source PS. Ces expériences se rapportent aux images des fronts d'ondes présentées sur les Fig. 3.3, 3.4 et 3.5. Nous rappelons que l'énergie s'exprime en $J.s^{-2}$ car nous travaillons en vitesse et non en déplacement (et en considérant la masse volumique en 2D comme une masse surfacique). Le cas homogène permet de décrypter les différentes étapes de la simulation. L'énergie croît fortement puis décroît par paliers. C'est sur le second palier, à $t = 5s$ environ, que l'efficacité d'une CLA peut être mesurée. Sur les trois configurations i.e. homogène, bi-couches bande et coin, les différences entre les trois types de source sont les mêmes que pour les cas isotrope et VTI. Une simulation avec une source PS a une courbe d'énergie correspondant plutôt à l'absorption de l'onde S et ne permettant pas de distinguer les petites réflexions de l'onde P.

Comparaison avec la CLA isotrope

En utilisant une CLA isotrope dans un milieu TTI, nous avons observé à la Fig. 3.6 que les réflexions d'ondes P étaient très fortes. L'étude de l'énergie sur le cas homogène de cette expérience est présentée sur la Fig. 3.10. Les courbes de la source en onde P confirment les observations, les réflexions de l'onde P transformée en onde P, palier à 0.1, mais aussi transformée en onde S, avec un palier à 0.01, sont plus fortes avec une CLA isotrope qu'avec une CLA TTI où nous observons un palier en dessous de 0.01. En comparaison avec les simulations dotées d'une source en onde PS ou en onde S, la première réflexion (onde P transformée en onde P) se produit alors que l'onde S n'est pas encore totalement absorbée et la deuxième réflexion (onde P transformée en onde S) a une intensité plus faible que les réflexions de l'onde S (transformée en ondes P et S). Ces réflexions sont visibles sur les images des fronts d'ondes mais ne se détectent que sur les courbes d'énergie de la simulation avec une source en onde P.

Non-elliptique

Les énergies correspondant aux expériences avec un milieu TTI non-elliptique (Fig. 3.7) sont présentées Fig. 3.11 pour une source élastique. Le comportement est le même que pour un milieu TTI elliptique (Fig. 3.9). La comparaison avec le cas homogène pour une CLA isotrope ou VTI est présentée à la Fig. 3.12, correspondant à l'expérience de la Fig. 3.8 où la CLA isotrope est utilisée dans un milieu TTI non-elliptique. Comme la source est élastique, les réflexions de l'onde P visibles sur les images ne sont pas évaluables par les courbes d'énergie.

Simulation en temps long

Nous avons prouvé la stabilité du problème mixte discret modulo une condition CFL. Néanmoins, nous savons que l'utilisation de conditions aux limites instables peut faire que la solution numérique se mette à croître de façon exponentielle après un temps long de simulation, c'est-à-dire bien après les réflexions des ondes. Nous proposons donc ici de mesurer l'énergie discrète pour un temps long de simulation. La Fig. 3.13 regroupe les énergies discrètes mesurées sur des cas TTI elliptique et TTI non-elliptique pour des configurations homogène, bi-couches bande et coin avec une source élastique. Une fois la précision machine atteinte, la mesure de l'énergie ne montre aucune instabilité en temps long.

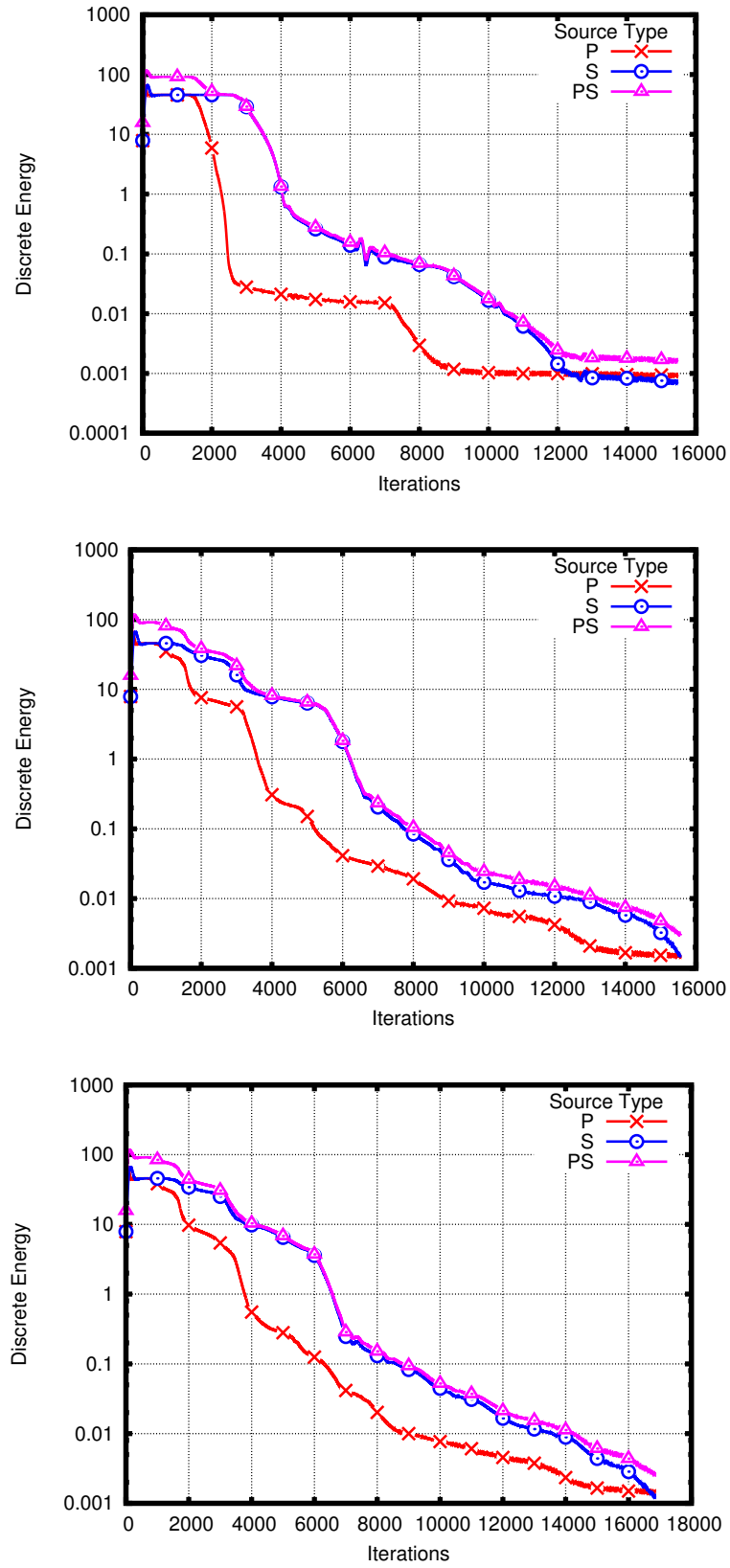


FIGURE 3.9 – Énergies discrètes pour les cas TTI 2D homogène (haut) et bi-couches, bande (centre) et coin (bas)

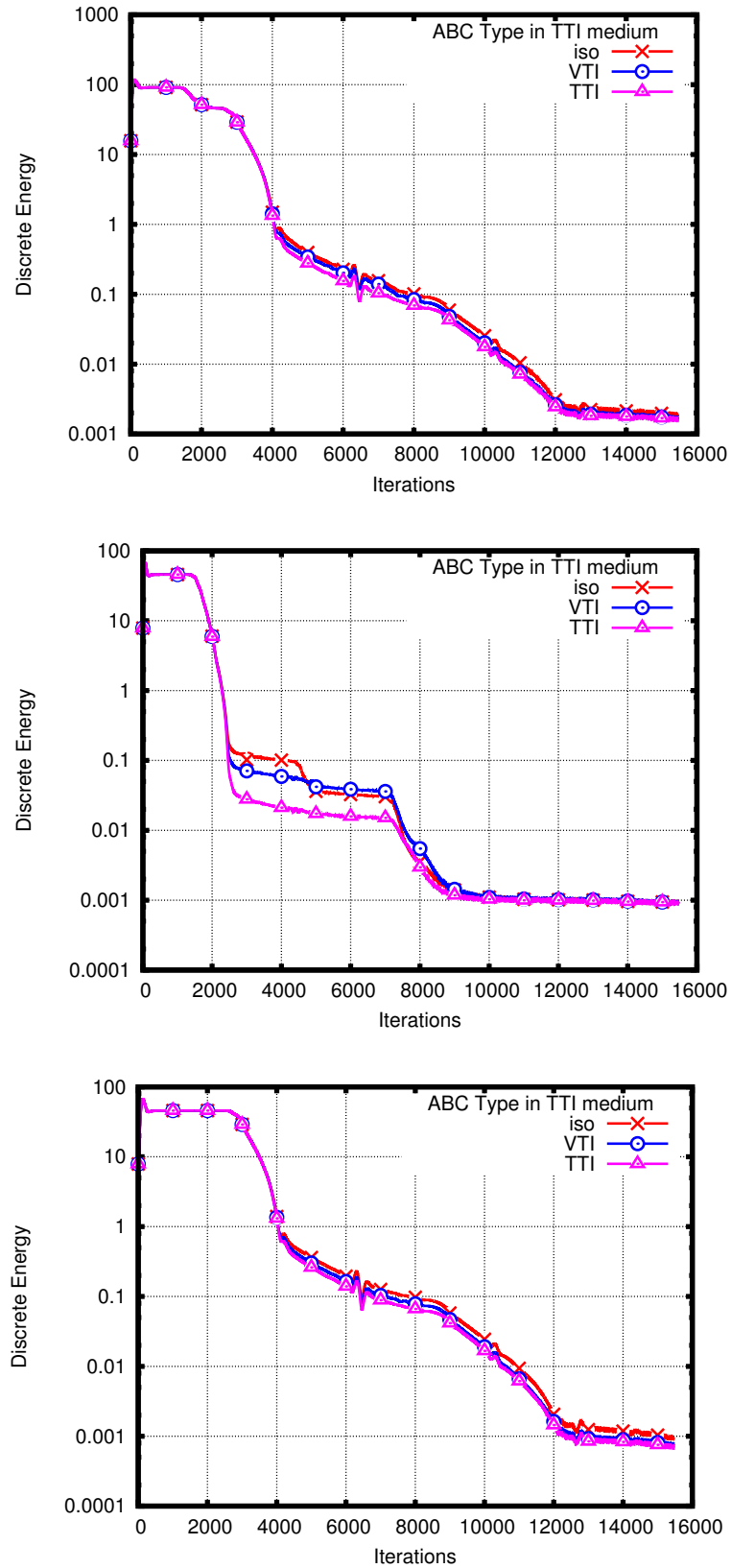


FIGURE 3.10 – Comparaison avec les CLA isotrope et VTI dans un milieu homogène TTI 2D, avec une source PS (haut), une source P (centre) et une source S (bas)

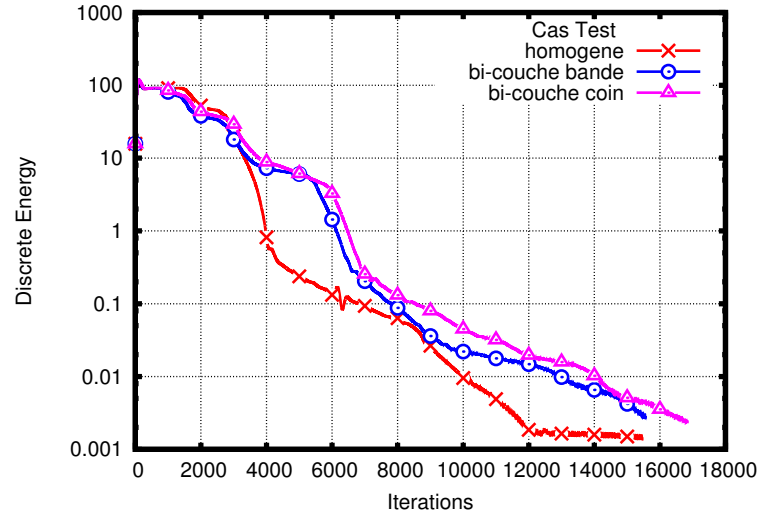


FIGURE 3.11 – Énergie discrète dans les cas TTI 2D non-elliptique

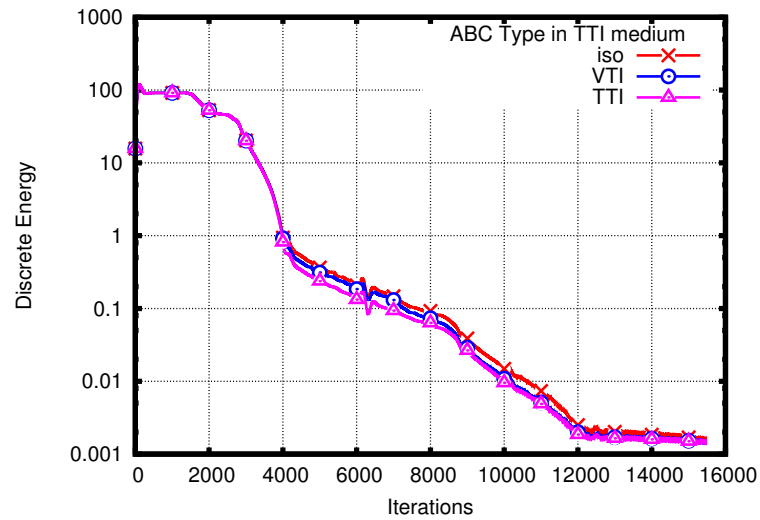


FIGURE 3.12 – Comparaison avec les CLA isotrope et VTI sur un cas TTI 2D non-elliptique

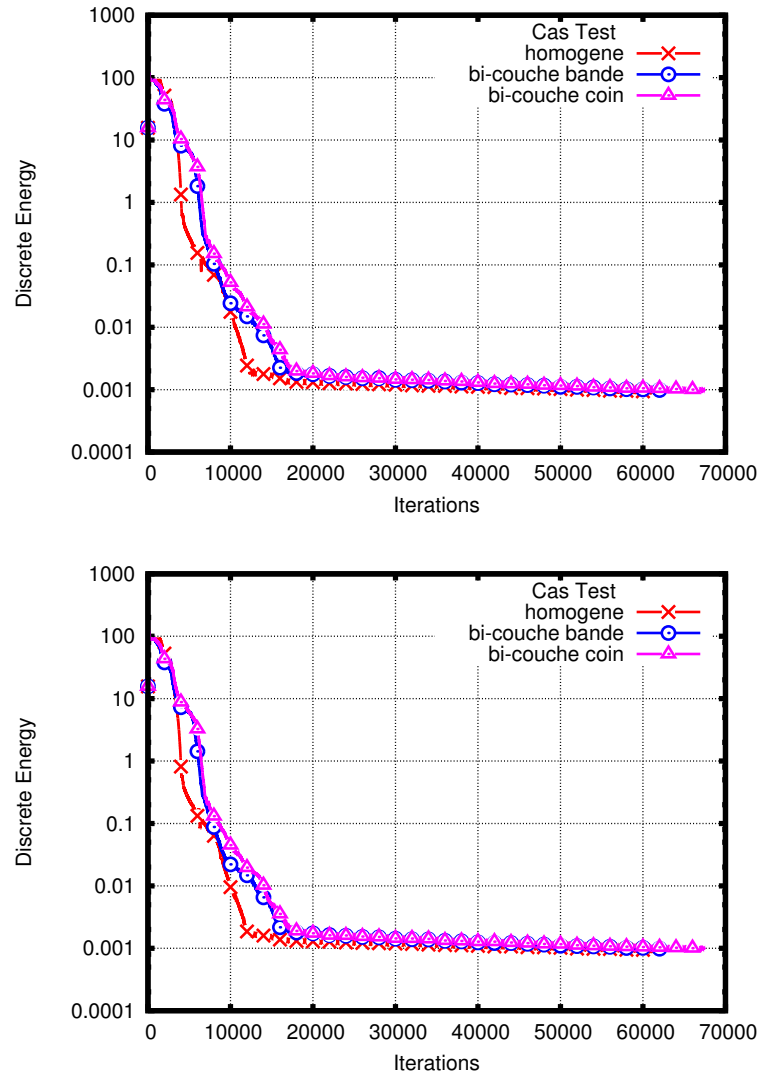


FIGURE 3.13 – Énergie discrète pour des temps long dans les cas TTI elliptique (haut) et TTI non-elliptique (bas) 2D

3.2 Coefficients de réflexion de la CLA TTI 2D

Au paragraphe précédent, nous avons étudié la stabilité de la CLA. Celle-ci est démontrée à l'aide d'une technique énergétique. Dans ce paragraphe, nous nous intéressons aux réflexions causées par le bord absorbant. Il s'agit donc d'analyser le comportement de la solution du problème mixte.

Nous considérons ici un domaine semi-infini $\{\Omega : x \in]-\infty, 0]\}$ avec une CLA en $\{x = 0\}$. Au bord du domaine, la condition aux limites absorbante réfléchit une partie des ondes. Une onde P produit deux réflexions d'ondes notées PP et PS, et une onde S produit deux réflexions d'ondes notées SP et SS. Le champ d'onde total est la somme de l'onde incidente et des ondes réfléchies :

$$\mathbf{u}^I = \mathbf{u}^{inc_I} + \mathbf{u}^{II} + \mathbf{u}^{IJ} \quad (3.54)$$

avec (I, J) représentant (P, S) pour une onde incidente P et (S, P) pour une onde incidente S.

Soit la matrice de l'équation de dispersion du système de l'élastodynamique en dimension deux :

$$E_{TTI,2D} = \begin{pmatrix} C_{11}k_x^2 + C_{55}k_z^2 + 2C_{15}k_xk_z & C_{15}k_x^2 + (C_{55} + C_{13})k_xk_z + C_{35}k_z^2 \\ C_{15}k_x^2 + (C_{55} + C_{13})k_xk_z + C_{35}k_z^2 & C_{55}k_x^2 + C_{33}k_z^2 + 2C_{35}k_xk_z \end{pmatrix} \quad (3.55)$$

où C est la matrice d'élasticité TTI 2D (3.1).

Sous l'hypothèse d'ellipticité ($\kappa = \sqrt{1 + 2\varepsilon} = \sqrt{1 + 2\delta}$), les vecteurs propres de la matrice $E_{TTI,2D}$ sont donnés par :

$$\begin{cases} \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta)ik_x + \cos \theta \sin \theta (X - Z)ik_z \\ \cos \theta \sin \theta (X - Z)ik_x + (Z \cos^2 \theta + X \sin^2 \theta)ik_z \end{pmatrix}, \\ \begin{pmatrix} -\cos \theta \sin \theta (X - Z)ik_x - (Z \cos^2 \theta + X \sin^2 \theta)ik_z \\ (X \cos^2 \theta + Z \sin^2 \theta)ik_x + \cos \theta \sin \theta (X - Z)ik_z \end{pmatrix} \end{cases} \quad (3.56)$$

avec $X = \sqrt{\kappa^2 V_p^2 + V_s^2}$ et $Z = \sqrt{V_p^2 + V_s^2}$. L'angle θ étant le coefficient de Thomsen de rotation TTI.

Nous notons A_{II} et A_{IJ} les coefficients de réflexion. Ils interviennent de la façon suivante :

$$\begin{cases} \mathbf{u}^{inc} &= \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta)ik_x + \cos \theta \sin \theta (X - Y)ik_z \\ \cos \theta \sin \theta (X - Y)ik_x + (Z \cos^2 \theta + X \sin^2 \theta)ik_z \end{pmatrix} p^{inc} \\ \mathbf{u}^{rp} &= \begin{pmatrix} -(X \cos^2 \theta + Z \sin^2 \theta)ik_x^p + \cos \theta \sin \theta (X - Y)ik_z \\ -\cos \theta \sin \theta (X - Y)ik_x^p + (Z \cos^2 \theta + X \sin^2 \theta)ik_z \end{pmatrix} A_{pp} p^{rp} \\ \mathbf{u}^{rs} &= \begin{pmatrix} \cos \theta \sin \theta (X - Y)ik_x^s - (Z \cos^2 \theta + X \sin^2 \theta)ik_z \\ -(X \cos^2 \theta + Z \sin^2 \theta)ik_x^s + \cos \theta \sin \theta (X - Y)ik_z \end{pmatrix} A_{ps} p^{rs} \end{cases} \quad (3.57)$$

3.2. Coefficients de réflexion de la CLA TTI 2D

avec

$$p^{inc} = e^{i\omega t + ik_x x + ik_z z}, \quad p^I = e^{i\omega t + ik_x^I x + ik_z^I z}, \quad p^J = e^{i\omega t + ik_x^J x + ik_z^J z} \quad (3.58)$$

pour un bord de normale extérieure colinéaire à \mathbf{e}_x , sur lequel nous avons $k_z^J = k_z^I = k_z$.

Les coefficients de réflexion de l'onde S peuvent s'exprimer, comme en VTI, en fonction d'un paramètre θ_0 qui correspond à l'angle d'incidence de l'onde sur le bord absorbant. Ce paramètre détermine la valeur des rapports $\frac{k_x}{w}$. Pour l'onde P, comme le front d'onde est incliné, il est plus simple de se référer directement à l'un de ses rapports. Nous choisissons $\frac{k_z}{w}$ et même plus directement k_z . La Fig. 3.14 représente ces rapports pour une onde incidente P ou S. Le rapport $\frac{k_z}{w}$ peut toujours se déterminer par trigonométrie pour l'onde S : $\frac{k_z}{w} = -\frac{\sin \theta_0}{V_s}$.

Contrairement au cas VTI, les rapports $\frac{k_x}{w}$, $\frac{k_x^I}{w}$ et $\frac{k_x^J}{w}$ ne se déduisent plus par symétrie et par la loi de Snell-Descartes. Ils se calculent maintenant *via* les équations du cercle et de l'ellipse

$$\xi_1 k_x^2 + \xi_2 k_x k_z + \xi_3 k_z^2 = 1 \quad (3.59)$$

avec

$$\begin{cases} \xi_1 &= \rho V_p^2 (\kappa^2 \cos^2 \theta + \sin^2 \theta) \\ \xi_2 &= 2 \cos \theta \sin \theta \rho V_p^2 (\kappa^2 - 1) \\ \xi_3 &= \rho V_p^2 (\kappa^2 \sin^2 \theta + \cos^2 \theta). \end{cases} \quad (3.60)$$

Ainsi, il est possible d'écrire, à k_z fixé :

$$\left\{ \begin{array}{l} \frac{k_x}{w} = \frac{-\xi_2 k_z + \sqrt{k_z^2 \xi_2^2 - 4\xi_1(\xi_3 k_z^2 - 1)}}{2\xi_1}, \quad \frac{k_x^P}{w} = \frac{-\xi_2 k_z - \sqrt{k_z^2 \xi_2^2 - 4\xi_1(\xi_3 k_z^2 - 1)}}{2\xi_1}, \\ \text{et } \frac{k_x^S}{w} = -\sqrt{\frac{1}{V_s^2} - k_z^2}, \text{ pour l'onde P,} \\ \frac{k_x}{w} = \frac{\cos \theta_0}{V_s}, \quad \frac{k_x^S}{w} = -\frac{\cos \theta_0}{V_s}, \quad \frac{k_x^P}{w} = \frac{-\xi_2 k_z - \sqrt{k_z^2 \xi_2^2 - 4\xi_1(\xi_3 k_z^2 - 1)}}{2\xi_1}, \text{ pour l'onde S.} \end{array} \right. \quad (3.61)$$

Les coefficients de réflexion ne sont plus symétriques en k_z ou θ_0 comme en VTI. Pour l'onde S, le paramètre angulaire θ_0 parcourt l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Pour l'onde P, les extrémités de l'intervalle de k_z correspondent aux points de la courbe de lenteur où les vitesses de groupe sont verticales, c'est-à-dire aux points où la tangente est horizontale (voir Fig. 3.15). Ces points

sont la double racine de l'équation de l'ellipse à k_x fixé : $k_z \in \left[-\frac{2\sqrt{\xi_1}}{\sqrt{-\xi_2^2 + 4\xi_1\xi_3}}, \frac{2\sqrt{\xi_1}}{\sqrt{-\xi_2^2 + 4\xi_1\xi_3}} \right]$.

Remarque 3.6 *Au-delà d'un certain angle critique, et comme physiquement $V_s < V_p$, le rapport $\frac{k_x^P}{w}$ devient complexe pour l'onde S. De plus, pour les deux ondes, au-delà d'un autre angle critique, les rapports $\frac{k_x}{w}$ et $\frac{k_x^P}{w}$ sont tous les deux du même signe. Nous retrouvons également cette propriété sur la Fig. 3.14.*

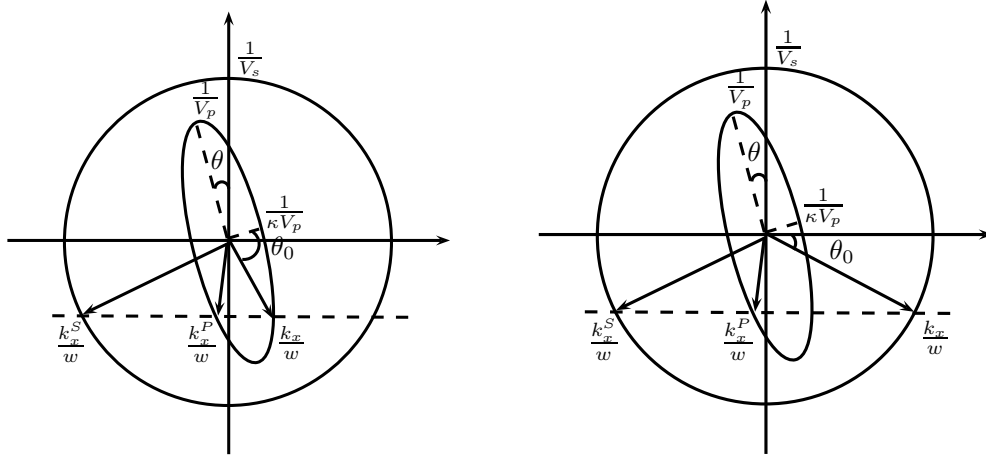


FIGURE 3.14 – Schéma des angles intervenant dans la réflexion de l'onde P (gauche) et de l'onde S (droite), en TTI

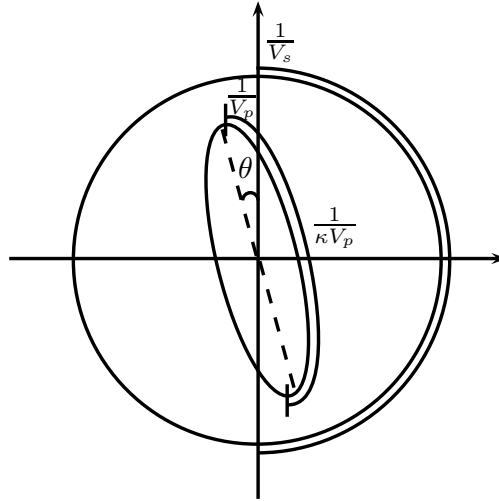


FIGURE 3.15 – Domaine d'étude des coefficients de réflexion de l'onde P et de l'onde S en TTI

Sur un bord de normale extérieure colinéaire à \mathbf{e}_x , le champ total \mathbf{u} vérifie la CLA (3.41) :

$$\left\{ \begin{array}{l} C_{11}\partial_x u_x + C_{13}\partial_z u_z + C_{15}\partial_x u_z + C_{15}\partial_z u_x \\ \quad = \rho V_p \frac{\kappa \cos^2 \theta + \sin^2 \theta}{(\kappa^2 \cos^2 \theta + \sin^2 \theta)^{1/2}} \partial_t [(\kappa \cos^2 \theta + \sin^2 \theta) u_x + (\kappa - 1) \cos \theta \sin \theta u_z] \\ C_{15}\partial_x u_x + C_{35}\partial_z u_z + C_{55}\partial_x u_z + C_{55}\partial_z u_x \\ \quad = \rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta}{(\kappa^2 \cos^2 \theta + \sin^2 \theta)^{1/2}} \partial_t [(\kappa \cos^2 \theta + \sin^2 \theta) u_x + (\kappa - 1) \cos \theta \sin \theta u_z] + \rho V_s \partial_t u_z \end{array} \right. \quad (3.62)$$

Comme au chapitre précédent, la suite du processus consiste à utiliser (3.57) dans la

3.2. Coefficients de réflexion de la CLA TTI 2D

première et la dernière équation de (3.62), pour l'onde P ou l'onde S. Ensuite, le fait de se placer sur la frontière, c'est-à-dire en $\{x = 0\}$, permet de simplifier les exponentielles (3.58) des termes p^* qui sont toutes égales. Il en résulte un système matriciel sur l'onde P ou l'onde S. Malheureusement, les solutions de ce système sont très complexes et le logiciel MapleTM est dans l'incapacité d'en fournir un développement limité. Il reste néanmoins la possibilité de calculer ces coefficients pour des paramètres physiques donnés, voir annexe A.6.

La Fig. 3.16 représente les coefficients de réflexion pour un milieu TTI elliptique, avec la CLA TTI, la CLA VTI et la CLA isotrope. Les courbes ne sont plus symétriques comme en VTI mais présentent tout de même des similitudes avec les courbes VTI (Fig. 2.19). Il existe au moins un point, hors extrémités, sur chaque courbe de CLA TTI où les réflexions sont nulles. Aux extrémités, ce qui correspond à la notion d'incidence rasante en TTI, les coefficients A_{PP} et A_{SS} génèrent une réflexion totale. Le coefficient A_{PS} est assez faible tandis que le coefficient A_{SP} est plus important. De plus, A_{PS} et A_{SP} sont nuls aux extrémités mais présentent un pic tout proche. D'ailleurs, la courbe A_{SP} de la CLA TTI dépasse la valeur de 1 dans cette zone.

La comparaison entre l'utilisation d'une CLA TTI et d'une CLA VTI ou isotrope dans un milieu TTI met en évidence la supériorité de la CLA TTI au centre des intervalles d'observation, correspondant à la notion d'incidence normale en TTI. Ceci s'observe par des coefficients moins élevés et surtout par les coefficients A_{PP} et A_{PS} des CLA isotrope et VTI qui sont plus importants dans cette zone. Les choix d'approximation et de simplification ont été faits dans ce sens. Par contre, contrairement au cas d'un milieu VTI, les courbes des CLA isotrope et VTI ne sont pas toujours en-dessous des courbes de la CLA TTI dans les zones proches des extrémités.

Remarque 3.7 *Ces résultats sont à corrélérer avec les images des fronts d'ondes de la CLA TTI (Fig. 3.3), de la CLA isotrope dans un milieu TTI (Fig. 3.6) et des courbes d'énergies discrètes (Fig. 3.10).*

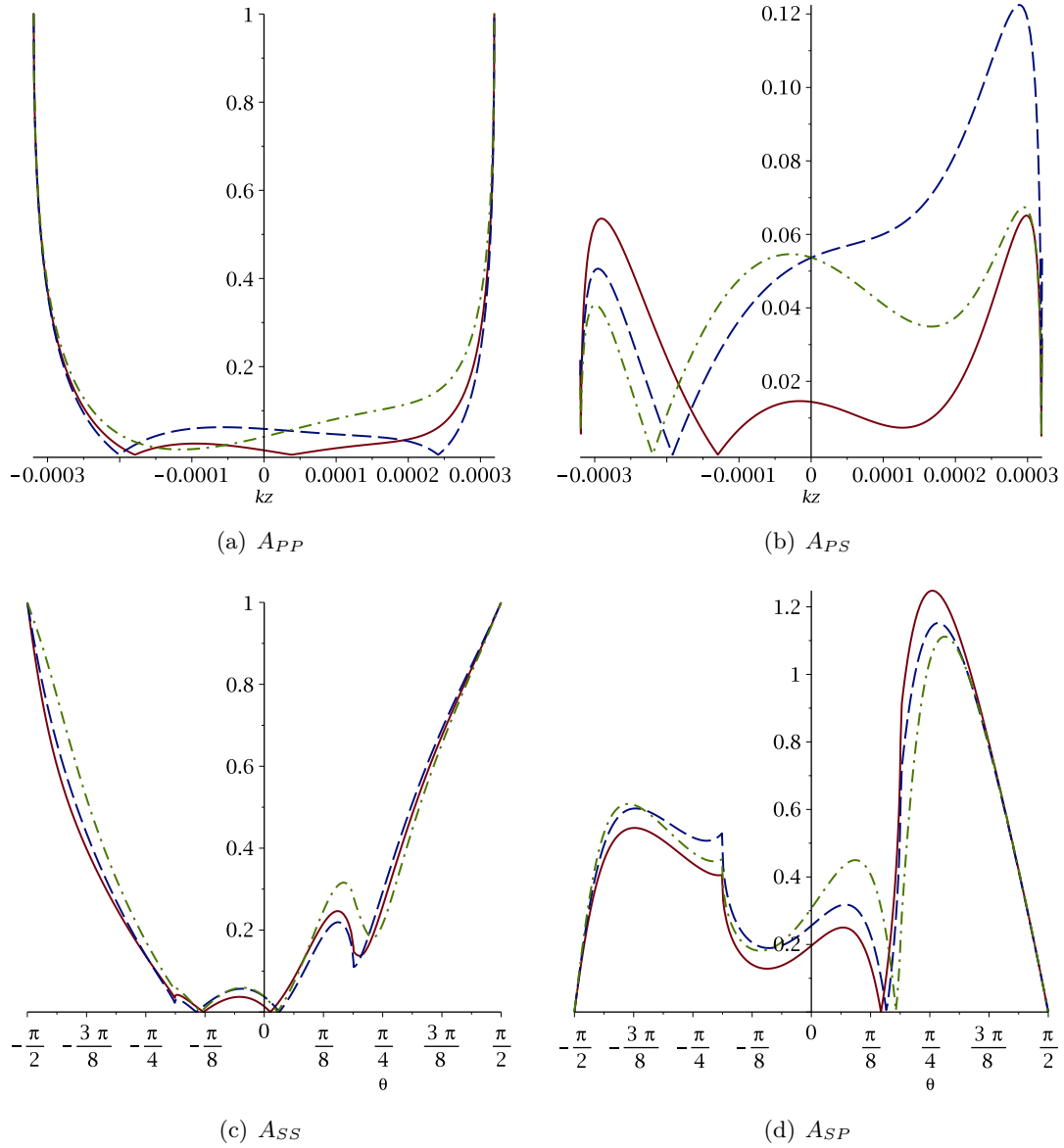


FIGURE 3.16 – Coefficients de réflexion de la CLA TTI 2D pour $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1750 \text{ m.s}^{-1}$, $\varepsilon = 0.25$ et $\theta = 30^\circ$, en fonction de θ_0 ou k_z , en rouge trait plein, comparaison avec l'utilisation d'une CLA VTI, en bleu trait pointillé et d'une CLA isotrope, en vert trait semi-pointillé, dans le même milieu TTI elliptique

3.3 Extension au cas 3D

3.3.1 Construction d'une CLA pour un matériau TTI 3D

Nous allons suivre la même démarche qu'en dimension deux, détaillée dans la section 3.1. Ici encore, nous nous plaçons dans un contexte d'anisotropie elliptique, c'est-à-dire en fixant $\delta = \varepsilon$. Cette hypothèse se justifie après avoir observé que le coefficient de Thomsen δ n'apparaissait pas dans la CLA d'ordre faible VTI 2D (2.53) et qu'un milieu TTI 2D peut se décrire

comme une rotation d'un milieu VTI 2D. En dimension trois, cette propriété est toujours vraie et nous allons l'exploiter encore une fois.

En dimension trois, un matériau élastique laisse se propager trois ondes : une onde P et deux ondes S. Dans le cas d'un matériau isotrope, les deux ondes S sont confondues mais dans le cas d'un matériau anisotrope, les deux ondes S sont indépendantes. Pour l'anisotropie de type transverse isotrope, les deux ondes S sont différenciées par le coefficient de Thomsen γ défini dans (1.22).

Surfaces de lenteur

Les surfaces de lenteur sont très commodes pour représenter l'anisotropie d'un milieu. La Fig. 3.17 les représentent sur un cas isotrope. En dimension trois, les surfaces de lenteur sont des sphères pour les cas isotropes. Les Fig. 3.18 et 3.19 concernent les cas VTI et TTI respectivement, en différenciant les configurations elliptique et non-elliptique. Les courbes ont été obtenues *via* des calculs réalisés à partir du logiciel MapleTM (voir annexe A.7). Comme en dimension deux, le fait de considérer un cas d'anisotropie TI elliptique conduit à des formes simples à paramétrer. La surface de lenteur de l'onde P forme un ellipsoïde de révolution (aplatis dans le plan (x, y) en VTI). Celles des ondes S forment une sphère et un ellipsoïde de révolution (également aplati dans le plan (x, y) en VTI).

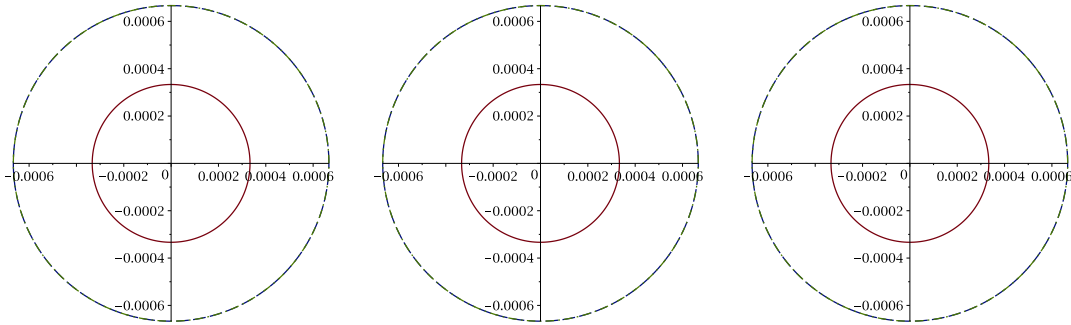
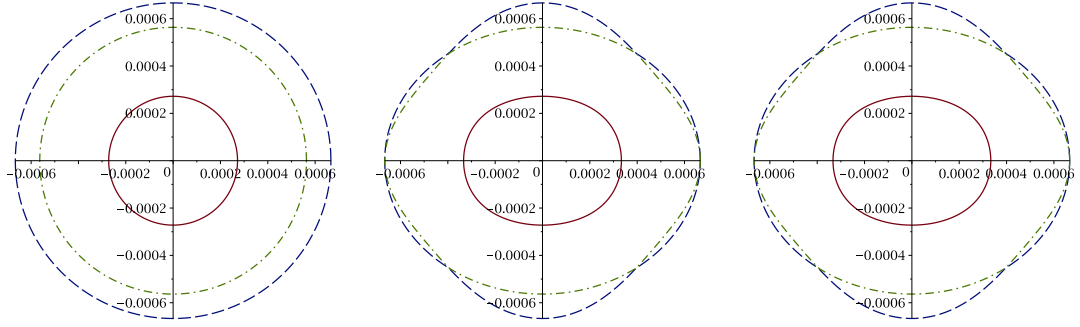
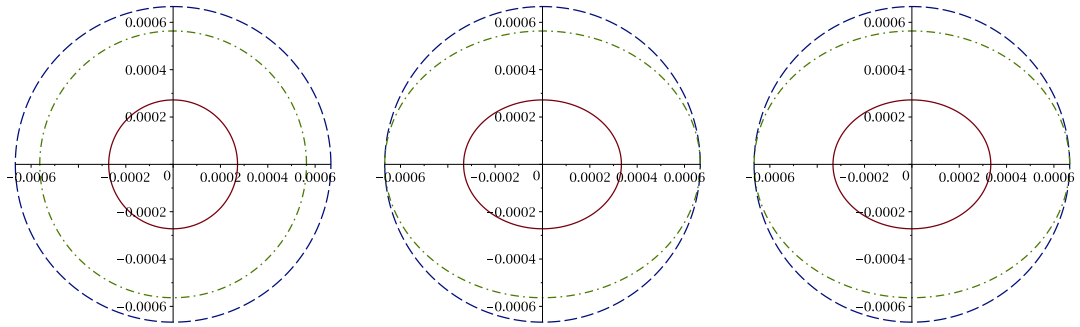


FIGURE 3.17 – Surfaces de lenteur élastiques isotropes : plan (x, y) (à gauche), plan (x, z) (au milieu) et plan (y, z) (à droite) avec $\rho = 1 \text{ kg.m}^{-3}$, $V_p = 3000 \text{ m.s}^{-1}$ et $V_s = 1800 \text{ m.s}^{-1}$, pour les ondes P en rouge trait plein, pour les ondes S en bleu trait pointillé et vert trait semi-pointillé

La méthode de construction d'une CLA TTI pour la 3D s'appuie sur la formulation d'un changement de variables basé sur des géométries simples comme la sphère ou l'ellipsoïde de révolution. Ces géométries sont dictées par les formes connues des surfaces de lenteur. Nous choisissons comme référence le cas isotrope et les solutions du problème correspondant sont repérées grâce au symbole étoile. Soit $(\mathbf{v}^*, \underline{\sigma}^*)$ la solution du problème de propagation d'ondes élastiques isotropes et $(\mathbf{v}, \underline{\sigma})$ celle du problème anisotrope TTI elliptique. Nous allons exprimer la relation entre ces deux solutions et nous l'utiliserons ensuite pour déduire une CLA TTI d'une CLA isotrope. Les géométries simples des surfaces de lenteur fournissent un premier changement de variables entre les modes de propagation des systèmes isotrope (k_x^*, k_y^*, k_z^*) et TTI elliptique (k_x, k_y, k_z) . Il reste à déterminer, par identification, les changements de



(a) configuration non-elliptique



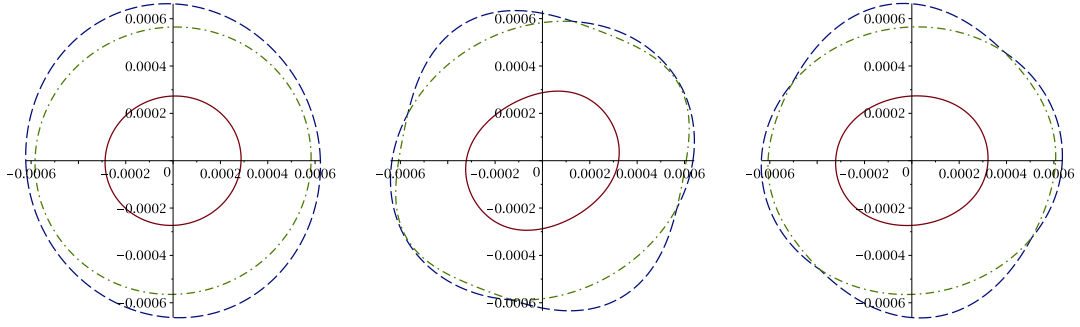
(b) configuration elliptique ($\delta = \varepsilon$)

FIGURE 3.18 – Surfaces de vitesse élastiques VTI : plan (x, y) (à gauche), plan (y, z) (au milieu) et plan (x, z) (à droite) avec $\rho = 1\text{kg.m}^{-3}$, $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1800 \text{ m.s}^{-1}$, $\varepsilon = 0.25$, $\delta = 0.10$ et $\gamma = 0.20$, pour les ondes P en rouge trait plein, pour les ondes S en bleu trait pointillé et vert trait semi-pointillé

variables sur les vitesses \mathbf{v} et \mathbf{v}^* et les tenseurs des contraintes $\underline{\underline{\sigma}}$ et $\underline{\underline{\sigma}}^*$.

CLA TTI elliptique pour les ondes P

Comme en dimension deux, les surfaces de vitesse des ondes P et des ondes S sont découplées, ne permettant pas de formuler un changement de variables sur les trois formes géométriques en même temps (voir Fig. 3.17, 3.18 et 3.19). Nous allons donc à nouveau découpler les problèmes pour construire des CLA TTI pseudo-acoustiques pour les ondes P et les ondes S. Lorsque $\gamma = 0$, les ondes S sont confondues, permettant de déduire directement la CLA d'ordre faible TTI elliptique pour les ondes S à partir de la configuration isotrope. La démarche est alors similaire à ce que nous avons fait en 2D. Lorsque $\gamma \neq 0$, la surface de vitesse d'une des ondes S est toujours une sphère, mais celle de la seconde est un ellipsoïde. Cet ellipsoïde admet les mêmes axes principaux que la surface de vitesse de l'onde P et seuls les rayons différent. Cependant, le processus de construction de la CLA pour cette onde S s'avère plus compliqué que pour l'onde P car il nécessite de découpler les deux ondes S. La suite des développements se concentre sur la construction d'une CLA d'ordre faible TTI elliptique pour les ondes P, c'est-à-dire en fixant temporairement $V_s = 0$ (ce qui ne fait plus intervenir γ).



(a) configuration non-elliptique

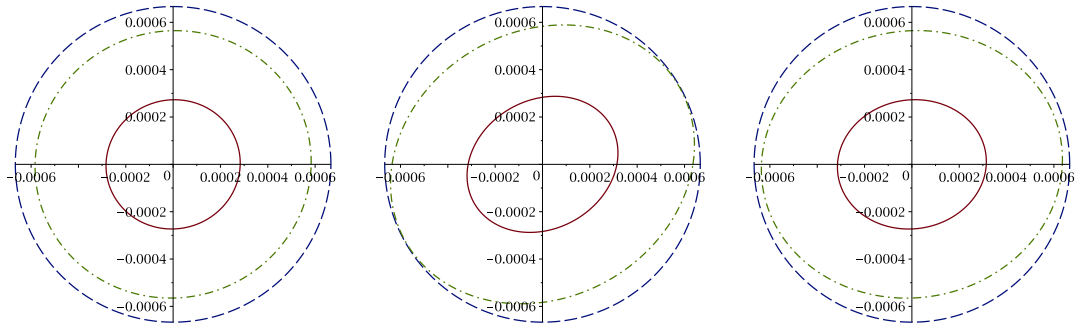

 (b) configuration elliptique ($\delta = \varepsilon$)

FIGURE 3.19 – Surfaces de lenteur élastiques TTI : plan (x, y) (à gauche), plan (y, z) (au milieu) et plan (x, z) (à droite) avec $\rho = 1 \text{ kg.m}^{-3}$, $V_p = 3000 \text{ m.s}^{-1}$, $V_s = 1800 \text{ m.s}^{-1}$, $\varepsilon = 0.25$, $\delta = 0.10$, $\gamma = 0.20$, $\theta = 30^\circ$ et $\phi = 15^\circ$, pour les ondes P en rouge trait plein, pour les ondes S en bleu trait pointillé et vert trait semi-pointillé

Étape 1 : changement de variables sur les modes de propagation à l'aide des formes géométriques sur les surfaces de lenteur. Il s'agit de lier le vecteur d'onde $\mathbf{k} = (k_x, k_y, k_z)$ du milieu TTI elliptique au vecteur d'ondes $\mathbf{k}^* = (k_x^*, k_y^*, k_z^*)$ du milieu isotrope. Nous proposons le changement de variables suivant :

$$\begin{cases} k_x &= \mu_{11}k_x^* + \mu_{12}k_y^* + \mu_{13}k_z^* \\ k_y &= \mu_{21}k_x^* + \mu_{22}k_y^* + \mu_{23}k_z^* \\ k_z &= \mu_{31}k_x^* + \mu_{32}k_y^* + \mu_{33}k_z^* \end{cases} \quad (3.63)$$

où $\{\mu_{ij}\}$, $1 \leq i, j \leq 3$ désignent les coefficients à déterminer.

En rappelant que les composantes k_x , k_y et k_z du vecteur d'onde peuvent s'interpréter comme les symboles des dérivées ∂_x , ∂_y et ∂_z , ce changement de variables peut être réinterprété comme un changement de variables dans le plan (x, y, z) de la forme :

$$\begin{cases} x^* &= \mu_{11}x + \mu_{21}y + \mu_{31}z \\ y^* &= \mu_{12}x + \mu_{22}y + \mu_{32}z \\ z^* &= \mu_{13}x + \mu_{23}y + \mu_{33}z \end{cases} \quad (3.64)$$

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

Nous avons choisi de construire une CLA posée sur la frontière $\{x = 0\}$ de normale extérieure \mathbf{e}_x . La frontière artificielle étant une donnée du problème, le changement de variables ne doit pas la modifier et donc préserver la droite d'équation $x = 0$. Or, appliquer une rotation d'angles TTI θ et ϕ n'est pas compatible avec cette contrainte. Nous devons donc construire une bijection entre une sphère et un ellipsoïde et conservant la droite $x = 0$. Pour cela, il faut éliminer les coefficients agissant sur les dérivées en y et z . Nous imposons donc $\mu_{21} = \mu_{31} = 0$. Il reste alors un degré de liberté à fixer qui consiste à choisir μ_{22} , μ_{23} , μ_{32} ou μ_{33} , dû à l'invariance par rotation autour de l'axe TTI. Quel que soit le paramètre fixé, nous obtiendrons la même CLA et le choix $\mu_{32} = 0$ est celui qui nous semble simplifier le plus les calculs.

En résumé, le changement de variables à déterminer est de la forme :

$$\begin{cases} k_x &= \mu_{11}k_x^* + \mu_{12}k_y^* + \mu_{13}k_z^* \\ k_y &= \mu_{22}k_y^* + \mu_{23}k_z^* \\ k_z &= \mu_{33}k_z^* \end{cases} \quad (3.65)$$

De plus, nous imposons que le déterminant du système (3.65) soit positif pour conserver l'orientation du repère, c'est-à-dire $\mu_{11}\mu_{22}\mu_{33} > 0$. Nous pouvons alors remarquer que nous cherchons à exprimer analytiquement la combinaison d'une dilatation d'axe \mathbf{e}_x de rapport μ_{11} , d'une dilatation d'axe \mathbf{e}_y de rapport μ_{22} , d'une dilatation d'axe \mathbf{e}_z de rapport μ_{33} et d'une transvection telle que

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ 0 & \mu_{22} & \mu_{23} \\ 0 & 0 & \mu_{33} \end{pmatrix} = \begin{pmatrix} \mu_{11} & 0 & 0 \\ 0 & \mu_{22} & 0 \\ 0 & 0 & \mu_{33} \end{pmatrix} \begin{pmatrix} 1 & \frac{\mu_{12}}{\mu_{11}} & \frac{\mu_{13}}{\mu_{11}} \\ 0 & 1 & \frac{\mu_{23}}{\mu_{22}} \\ 0 & 0 & 1 \end{pmatrix} \quad (3.66)$$

Les formes géométriques des Fig. 3.17, 3.18 et 3.19 ont une représentation analytique connue que nous allons exploiter. Nous avons :

- les modes de propagation des ondes élastiques isotropes (k_x^*, k_y^*, k_z^*) satisfont la fonction quadratique :

$$\rho V_p^2 k_x^{*2} + \rho V_p^2 k_y^{*2} + \rho V_p^2 k_z^{*2} = 1 \quad (3.67)$$

- les modes de propagation des ondes élastiques TTI elliptiques (k_x, k_y, k_z) satisfont l'équation générique :

$$\xi_1 k_x^2 + \xi_2 k_x k_y + \xi_3 k_x k_z + \xi_4 k_y^2 + \xi_5 k_y k_z + \xi_6 k_z^2 = 1 \quad (3.68)$$

avec :

$$\begin{cases} \xi_1 &= \rho V_p^2 (\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi) \\ \xi_2 &= 2 \cos \phi \sin \phi \sin \theta \rho V_p^2 (\kappa^2 - 1) \\ \xi_3 &= 2 \sin \theta \cos \phi \cos \theta \rho V_p^2 (\kappa^2 - 1) \\ \xi_4 &= \rho V_p^2 (\kappa^2 \cos^2 \theta \sin^2 \phi + \kappa^2 \cos^2 \phi + \sin^2 \theta \sin^2 \phi) \\ \xi_5 &= 2 \sin \theta \sin \phi \cos \theta \rho V_p^2 (\kappa^2 - 1) \\ \xi_6 &= \rho V_p^2 (\kappa^2 \sin^2 \theta + \cos^2 \theta) \end{cases} \quad (3.69)$$

En substituant les équations du système (3.65) dans la relation (3.68), il vient :

$$\left\{ \begin{array}{l} 1 = \xi_1 (\mu_{11}^2 k_x^{\star 2} + \mu_{12}^2 k_y^{\star 2} + \mu_{13}^2 k_z^{\star 2} + 2\mu_{11}\mu_{12}k_x^{\star}k_y^{\star} + 2\mu_{11}\mu_{13}k_x^{\star}k_z^{\star} + 2\mu_{12}\mu_{13}k_y^{\star}k_z^{\star}) \\ + \xi_2 (\mu_{11}\mu_{22}k_x^{\star}k_y^{\star} + \mu_{11}\mu_{23}k_x^{\star}k_z^{\star} + \mu_{12}\mu_{22}k_y^{\star 2} + (\mu_{12}\mu_{23} + \mu_{13}\mu_{22})k_y^{\star}k_z^{\star} + \mu_{13}\mu_{23}k_z^{\star 2}) \\ + \xi_3 (\mu_{11}\mu_{33}k_x^{\star}k_z^{\star} + \mu_{12}\mu_{33}k_y^{\star}k_z^{\star} + \mu_{13}\mu_{33}k_z^{\star 2}) \\ + \xi_4 (\mu_{22}^2 k_y^{\star 2} + \mu_{23}^2 k_z^{\star 2} + 2\mu_{22}\mu_{23}k_y^{\star}k_z^{\star}) \\ + \xi_5 (\mu_{22}\mu_{33}k_y^{\star}k_z^{\star} + \mu_{23}\mu_{33}k_z^{\star 2}) \\ + \xi_6 \mu_{33}^2 k_z^{\star 2} \end{array} \right. \quad (3.70)$$

puis en factorisant :

$$\left\{ \begin{array}{l} 1 = k_x^{\star 2} \xi_1 \mu_{11}^2 \\ + k_x^{\star} k_y^{\star} (\xi_1 2\mu_{11}\mu_{12} + \xi_2 \mu_{11}\mu_{22}) \\ + k_x^{\star} k_z^{\star} (\xi_1 2\mu_{11}\mu_{13} + \xi_2 \mu_{11}\mu_{23} + \xi_3 \mu_{11}\mu_{33}) \\ + k_y^{\star 2} (\xi_1 \mu_{12}^2 + \xi_2 \mu_{12}\mu_{22} + \xi_4 \mu_{22}^2) \\ + k_y^{\star} k_z^{\star} (\xi_1 2\mu_{12}\mu_{13} + \xi_2 (\mu_{12}\mu_{23} + \mu_{13}\mu_{22}) + \xi_3 \mu_{12}\mu_{33} + \xi_4 2\mu_{22}\mu_{23} + \xi_5 \mu_{22}\mu_{33}) \\ + k_z^{\star 2} (\xi_1 \mu_{13}^2 + \xi_2 \mu_{13}\mu_{23} + \xi_3 \mu_{13}\mu_{33} + \xi_4 \mu_{23}^2 + \xi_5 \mu_{23}\mu_{33} + \xi_6 \mu_{33}^2) \end{array} \right. \quad (3.71)$$

Par identification de (3.71) avec l'équation de la sphère (3.67), en simplifiant par μ_{11} dans les équations 2 et 3 et en simplifiant l'équation 4 à l'aide de l'équation 2 et les équations 5 et 6 à l'aide de l'équation 3 puis en simplifiant par μ_{22} dans l'équation 5, il vient :

$$\left\{ \begin{array}{ll} \xi_1 \mu_{11}^2 & = \rho V_p^2 \\ \xi_1 2\mu_{12} + \xi_2 \mu_{22} & = 0 \\ \xi_1 2\mu_{13} + \xi_2 \mu_{23} + \xi_3 \mu_{33} & = 0 \\ -\xi_1 \mu_{12}^2 + \xi_4 \mu_{22}^2 & = \rho V_p^2 \\ \xi_2 \mu_{13} + \xi_4 2\mu_{23} + \xi_5 \mu_{33} & = 0 \\ -\xi_1 \mu_{13}^2 + \xi_4 \mu_{23}^2 + \xi_5 \mu_{23}\mu_{33} + \xi_6 \mu_{33}^2 & = \rho V_p^2 \end{array} \right. \quad (3.72)$$

Le changement de variables sur les modes de propagation (3.63) est maintenant déterminé :

$$\left\{ \begin{array}{l} \mu_{11} = \sqrt{1/\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} \\ \mu_{12} = (\kappa^2 - 1) \cos \phi \sin \phi \sin^2 \theta / \kappa \sqrt{(\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi)(\kappa^2 \cos^2 \theta + \sin^2 \theta)} \\ \mu_{13} = (\kappa^2 - 1) \cos \theta \sin \theta \cos \phi / \kappa \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \\ \mu_{22} = \sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} / \kappa \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \\ \mu_{23} = (\kappa^2 - 1) \cos \theta \sin \theta \sin \phi / \kappa \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \\ \mu_{33} = \frac{1}{\kappa} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \end{array} \right. \quad (3.73)$$

où nous avons fixé arbitrairement $\mu_{11} > 0$, $\mu_{22} > 0$ et $\mu_{33} > 0$, ce qui respecte bien $\mu_{11}\mu_{22}\mu_{33} > 0$.

Étape 2 : changement de variables sur la vitesse. Les notations $(v_x^{\star}, v_y^{\star}, v_z^{\star})$ et (v_x, v_y, v_z) représentent les composantes de la variable vitesse pour les systèmes isotrope et TTI elliptique

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

respectivement. En procédant comme à l'étape 1, nous proposons le changement de variables :

$$\begin{cases} v_x &= \alpha_{11}v_x^* + \alpha_{12}v_y^* + \alpha_{13}v_z^* \\ v_y &= \alpha_{21}v_x^* + \alpha_{22}v_y^* + \alpha_{23}v_z^* \\ v_z &= \alpha_{31}v_x^* + \alpha_{32}v_y^* + \alpha_{33}v_z^* \end{cases} \quad (3.74)$$

où les coefficients $\{\alpha_{ij}\}, 1 \leq i, j \leq 3$ désignent les coefficients à déterminer. Le problème se réécrit de façon matricielle comme :

$$\mathbf{v} = A\mathbf{v}^* \quad (3.75)$$

avec $\mathbf{v} = (v_x, v_y, v_z)$ et avec la matrice

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \quad (3.76)$$

En utilisant le système de l'élastodynamique au second ordre, nous faisons disparaître la variable auxiliaire du tenseur des contraintes, ce qui permet de ne plus considérer que la variable vitesse. Pour un matériau TTI elliptique, ce système s'écrit sous forme matricielle après transformée de Fourier en espace :

$$\rho \partial_t^2 \mathbf{v} = [C_{xx}k_x^2 + C_{xy}k_xk_y + C_{xz}k_xk_z + C_{yy}k_y^2 + C_{yz}k_yk_z + C_{zz}k_z^2]\mathbf{v} \quad (3.77)$$

avec

$$\begin{cases} C_{xx} = \begin{pmatrix} C_{11} & C_{16} & C_{15} \\ C_{16} & C_{66} & C_{56} \\ C_{15} & C_{56} & C_{55} \end{pmatrix}, C_{xy} = \begin{pmatrix} 2C_{16} & (C_{12} + C_{66}) & (C_{14} + C_{56}) \\ (C_{12} + C_{66}) & 2C_{26} & (C_{46} + C_{25}) \\ (C_{14} + C_{56}) & (C_{46} + C_{25}) & 2C_{45} \end{pmatrix} \\ C_{xz} = \begin{pmatrix} 2C_{15} & (C_{14} + C_{56}) & (C_{13} + C_{55}) \\ (C_{14} + C_{56}) & 2C_{46} & (C_{36} + C_{45}) \\ (C_{13} + C_{55}) & (C_{36} + C_{45}) & 2C_{35} \end{pmatrix}, C_{yy} = \begin{pmatrix} C_{66} & C_{26} & C_{46} \\ C_{26} & C_{22} & C_{24} \\ C_{46} & C_{24} & C_{44} \end{pmatrix}, \\ C_{yz} = \begin{pmatrix} 2C_{56} & (C_{46} + C_{25}) & (C_{36} + C_{45}) \\ (C_{46} + C_{25}) & 2C_{24} & (C_{23} + C_{44}) \\ (C_{36} + C_{45}) & (C_{23} + C_{44}) & 2C_{34} \end{pmatrix}, C_{zz} = \begin{pmatrix} C_{55} & C_{45} & C_{35} \\ C_{45} & C_{44} & C_{34} \\ C_{35} & C_{34} & C_{33} \end{pmatrix} \end{cases} \quad (3.78)$$

où C désigne la matrice d'élasticité TTI (1.26) calculée pour le cas elliptique (ie $\delta = \varepsilon$) et en fixant $V_s = 0$.

Pour un matériau isotrope et en fixant $V_s = 0$, le système de l'élastodynamique au second ordre s'écrit sous forme matricielle, après transformée de Fourier en espace :

$$\rho \partial_t^2 \mathbf{v}^* = \rho V_p^2 [I_{xx}k_x^{2*} + I_{xy}k_x^*k_y^* + I_{xz}k_x^*k_z^* + I_{yy}k_y^{2*} + I_{yz}k_y^*k_z^* + I_{zz}k_z^{2*}]\mathbf{v}^* \quad (3.79)$$

avec $\mathbf{v}^* = (v_x^*, v_y^*, v_z^*)$ et

$$\left\{ \begin{array}{l} I_{xx} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, I_{xy} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, I_{xz} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \\ I_{yy} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, I_{yz} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, I_{zz} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \right. \quad (3.80)$$

Nous poursuivons ensuite en exprimant (k_x, k_y, k_z) en fonction de (k_x^*, k_y^*, k_z^*) grâce à (3.73) puis en remplaçant (v_x, v_y, v_z) grâce à (3.74) et en injectant le tout dans le système (3.77). Il vient :

$$\left\{ \begin{array}{l} \rho \partial_t^2 \mathbf{A} \mathbf{v}^* = [\quad C_{xx}(\mu_{11}k_x^* + \mu_{12}k_y^* + \mu_{13}k_z^*)^2 + \\ \quad C_{xy}(\mu_{11}k_x^* + \mu_{12}k_y^* + \mu_{13}k_z^*)(\mu_{22}k_y^* + \mu_{23}k_z^*) + \\ \quad C_{xz}(\mu_{11}k_x^* + \mu_{12}k_y^* + \mu_{13}k_z^*)\mu_{33}k_z^* + \\ \quad C_{yy}(\mu_{22}k_y^* + \mu_{23}k_z^*)^2 + \\ \quad C_{yz}(\mu_{22}k_y^* + \mu_{23}k_z^*)\mu_{33}k_z^* + \\ \quad C_{zz}\mu_{33}^2k_z^{*2} \quad] \mathbf{A} \mathbf{v}^* \end{array} \right. \quad (3.81)$$

et qui se réécrit :

$$\left\{ \begin{array}{l} \rho \partial_t^2 \mathbf{A} \mathbf{v}^* = [\quad \mu_{11}^2 C_{xx} k_x^{*2} + \\ \quad (2\mu_{11}\mu_{12}C_{xx} + \mu_{11}\mu_{22}C_{xy})k_x k_y + \\ \quad (2\mu_{11}\mu_{13}C_{xx} + \mu_{11}\mu_{23}C_{xy} + \mu_{11}\mu_{33}C_{xz})k_x k_z + \\ \quad (\mu_{12}^2 C_{xx} + \mu_{12}\mu_{22}C_{xy} + \mu_{22}^2 C_{yy})k_y^{*2} + \\ \quad (2\mu_{12}\mu_{13}C_{xx} + (\mu_{12}\mu_{23} + \mu_{13}\mu_{22})C_{xy} + \mu_{12}\mu_{33}C_{xz} + 2\mu_{22}\mu_{23}C_{yy} + \mu_{22}\mu_{33}C_{yz})k_y k_z + \\ \quad (\mu_{13}^2 C_{xx} + \mu_{13}\mu_{23}C_{xy} + \mu_{13}\mu_{33}C_{xz} + \mu_{23}^2 C_{yy} + \mu_{23}\mu_{33}C_{yz} + \mu_{33}^2 C_{zz})k_z^{*2} \quad] \mathbf{A} \mathbf{v}^* \end{array} \right. \quad (3.82)$$

Par construction, le système (3.82) doit être équivalent au système isotrope (3.79). Nous procédons donc à l'identification terme à terme qui conduit à un système de six équations matricielles de la forme :

$$\left\{ \begin{array}{ll} A^{-1}\mu_{11}^2 C_{xx} A & = \rho V_p^2 I_{xx} \quad (3.83a) \\ A^{-1}(2\mu_{11}\mu_{12}C_{xx} + \mu_{11}\mu_{22}C_{xy}) A & = \rho V_p^2 I_{xy} \quad (3.83b) \\ A^{-1}(2\mu_{11}\mu_{13}C_{xx} + \mu_{11}\mu_{23}C_{xy} + \mu_{11}\mu_{33}C_{xz}) A & = \rho V_p^2 I_{xz} \quad (3.83c) \\ A^{-1}(\mu_{12}^2 C_{xx} + \mu_{12}\mu_{22}C_{xy} + \mu_{22}^2 C_{yy}) A & = \rho V_p^2 I_{yy} \quad (3.83d) \\ A^{-1}(2\mu_{12}\mu_{13}C_{xx} + (\mu_{12}\mu_{23} + \mu_{13}\mu_{22})C_{xy} \\ \quad + \mu_{12}\mu_{33}C_{xz} + 2\mu_{22}\mu_{23}C_{yy} + \mu_{22}\mu_{33}C_{yz}) A & = \rho V_p^2 I_{yz} \quad (3.83e) \\ A^{-1}(\mu_{13}^2 C_{xx} + \mu_{13}\mu_{23}C_{xy} + \mu_{13}\mu_{33}C_{xz} \\ \quad + \mu_{23}^2 C_{yy} + \mu_{23}\mu_{33}C_{yz} + \mu_{33}^2 C_{zz}) A & = \rho V_p^2 I_{zz} \quad (3.83f) \end{array} \right.$$

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

Remarquons que si A est solution de (3.83), alors λA est également solution, pour tout $\lambda \in \mathbb{R}^*$. Le changement de variables est donc déterminé à une constante multiplicative près. Nous introduisons :

$$\left\{ \begin{array}{l} a = -\frac{\sqrt{\rho}V_p}{x}(\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta)) \\ b = \frac{\sqrt{\rho}V_p}{x}(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta \\ c = \frac{\sqrt{\rho}V_p}{x}(\kappa - 1) \cos \theta \sin \theta \cos \phi \\ d = \frac{\sqrt{\rho}V_p}{xy}(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta \\ e = \frac{\sqrt{\rho}V_p}{xy}((\kappa - 1)(\cos^2 \phi + \kappa) \cos^2 \theta + \kappa \sin^2 \phi + \cos^2 \phi) \\ f = -\frac{\sqrt{\rho}V_p}{xy}(\kappa - 1) \kappa \cos \theta \sin \theta \sin \phi \\ g = \frac{\sqrt{\rho}V_p}{y}(\kappa - 1) \cos \theta \sin \theta \cos \phi \\ h = \frac{\sqrt{\rho}V_p}{y}(\kappa - 1) \cos \theta \sin \theta \sin \phi \\ i = \frac{\sqrt{\rho}V_p}{y}(\kappa \cos^2 \theta + \sin^2 \theta) \\ x = \sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} \\ y = \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta} \end{array} \right. \quad (3.84)$$

et les matrices intervenant dans les équations (3.83a), (3.83d) et (3.83f) peuvent se réécrire :

$$\left\{ \begin{array}{l} A^{-1} \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix} A = \rho V_p^2 I_{xx} \\ A^{-1} \begin{pmatrix} d^2 & de & df \\ de & e^2 & ef \\ df & ef & f^2 \end{pmatrix} A = \rho V_p^2 I_{yy} \\ A^{-1} \begin{pmatrix} g^2 & gh & gi \\ gh & h^2 & hi \\ gi & hi & i^2 \end{pmatrix} A = \rho V_p^2 I_{zz} \end{array} \right. \quad (3.85)$$

ce qui conduit aux relations :

$$\left\{ \begin{array}{l} \alpha_{21} = \alpha_{11}b/a \\ \alpha_{31} = \alpha_{11}c/a \\ \alpha_{12} = \alpha_{22}d/e \\ \alpha_{32} = \alpha_{22}f/e \\ \alpha_{13} = \alpha_{33}g/i \\ \alpha_{23} = \alpha_{33}h/i \end{array} \right. \quad (3.86)$$

En se servant de ces relations, les équations (3.83b), (3.83c) et (3.83e) donnent :

$$\left\{ \begin{array}{l} \alpha_{22} = -\alpha_{11}e \\ \alpha_{33} = -\alpha_{11}i \end{array} \right. \quad (3.87)$$

Il reste une indéterminée qui traduit le fait que le changement est déterminé à une constante près. Si nous fixons arbitrairement $\alpha_{11} = a$, la matrice de passage devient :

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} = \begin{pmatrix} a & -d & -g \\ b & -e & -h \\ c & -f & -i \end{pmatrix} \quad (3.88)$$

Le changement de variables sur les vitesses (3.74) s'écrit alors :

$$\left\{ \begin{array}{l} v_x = \sqrt{\rho} V_p \left[-\frac{\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} v_x^* \right. \\ \quad - \frac{(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_y^* \\ \quad \left. - \frac{(\kappa - 1) \cos \theta \sin \theta \cos \phi}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_z^* \right] \\ \\ v_y = \sqrt{\rho} V_p \left[\frac{(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} v_x^* \right. \\ \quad - \frac{(\kappa - 1) (\cos^2 \phi + \kappa) \cos^2 \theta + \kappa \sin^2 \phi + \cos^2 \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_y^* \\ \quad \left. - \frac{(\kappa - 1) \cos \theta \sin \theta \sin \phi}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_z^* \right] \\ \\ v_z = \sqrt{\rho} V_p \left[\frac{(\kappa - 1) \cos \theta \sin \theta \cos \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} v_x^* \right. \\ \quad + \frac{(\kappa - 1) \kappa \cos \theta \sin \theta \sin \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi} \sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_y^* \\ \quad \left. - \frac{(\kappa \cos^2 \theta + \sin^2 \theta)}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} v_z^* \right] \end{array} \right. \quad (3.89)$$

Étape 3 : changement de variables sur le tenseur des contraintes. Nous conservons les mêmes conventions que précédemment si bien que $\underline{\underline{\sigma}}^*$ et $\underline{\underline{\sigma}}$ représentent les tenseurs des contraintes des systèmes isotrope et TTI elliptique respectivement. De plus, dans le système de l'élastodynamique isotrope en ondes P (i.e. en fixant $V_s = 0$), le tenseur des contraintes se

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

résume à une seule variable p^* . Nous posons le changement de variables :

$$\begin{cases} \sigma_{xx} &= \beta_1 p^* \\ \sigma_{yy} &= \beta_2 p^* \\ \sigma_{zz} &= \beta_3 p^* \\ \sigma_{yz} &= \beta_4 p^* \\ \sigma_{xz} &= \beta_5 p^* \\ \sigma_{xy} &= \beta_6 p^* \end{cases} \quad (3.90)$$

où $\{\beta_j\}_{1 \leq j \leq 6}$ désignent les coefficients à déterminer.

En procédant comme dans l'étape 2, il suffit d'écrire le système de l'élastodynamique (1.5) pour un matériau isotrope en fixant $V_s = 0$ et en appliquant une transformée de Fourier en espace :

$$\begin{cases} \rho \partial_t v_x^* &= k_x^* p^* \\ \rho \partial_t v_y^* &= k_y^* p^* \\ \rho \partial_t v_z^* &= k_z^* p^* \\ \partial_t p^* &= \rho V_p^2 (k_x^* v_x^* + k_y^* v_z^* + k_z^* v_z^*) \end{cases} \quad (3.91)$$

puis pour un matériau TTI :

$$\begin{cases} \rho \partial_t v_x = k_x \sigma_{xx} + k_y \sigma_{xy} + k_z \sigma_{xz} \\ \rho \partial_t v_y = k_x \sigma_{xy} + k_y \sigma_{yy} + k_z \sigma_{yz} \\ \rho \partial_t v_z = k_x \sigma_{xz} + k_y \sigma_{yz} + k_z \sigma_{zz} \\ \partial_t \sigma_{xx} = C_{11} k_x v_x + C_{12} k_y v_y + C_{13} k_z v_z + C_{14} (k_x v_z + k_z v_x) + C_{15} (k_y v_z + k_z v_y) + C_{16} (k_x v_y + k_y v_x) \\ \partial_t \sigma_{yy} = C_{12} k_x v_x + C_{22} k_y v_y + C_{23} k_z v_z + C_{24} (k_x v_z + k_z v_x) + C_{25} (k_y v_z + k_z v_y) + C_{26} (k_x v_y + k_y v_x) \\ \partial_t \sigma_{zz} = C_{13} k_x v_x + C_{23} k_y v_y + C_{33} k_z v_z + C_{34} (k_x v_z + k_z v_x) + C_{35} (k_y v_z + k_z v_y) + C_{36} (k_x v_y + k_y v_x) \\ \partial_t \sigma_{yz} = C_{14} k_x v_x + C_{24} k_y v_y + C_{34} k_z v_z + C_{44} (k_x v_z + k_z v_x) + C_{45} (k_y v_z + k_z v_y) + C_{46} (k_x v_y + k_y v_x) \\ \partial_t \sigma_{xz} = C_{15} k_x v_x + C_{25} k_y v_y + C_{35} k_z v_z + C_{45} (k_x v_z + k_z v_x) + C_{55} (k_y v_z + k_z v_y) + C_{56} (k_x v_y + k_y v_x) \\ \partial_t \sigma_{xy} = C_{16} k_x v_x + C_{26} k_y v_y + C_{36} k_z v_z + C_{46} (k_x v_z + k_z v_x) + C_{56} (k_y v_z + k_z v_y) + C_{66} (k_x v_y + k_y v_x) \end{cases} \quad (3.92)$$

où C est la matrice d'élasticité TTI (3.1) exprimée pour un matériau TI elliptique (ie $\delta = \varepsilon$) en fixant $V_s = 0$.

Puis, en substituant les changements de variables : sur les modes de propagation (3.73) obtenu à l'étape 1, sur les vitesses (3.89) obtenu à l'étape 2 et sur les contraintes (3.90) dans le système TTI elliptique (3.92), nous obtenons un système en $\sigma_{xx}^*, \sigma_{zz}^*, \sigma_{xz}^*$ qui doit être équivalent au système isotrope (3.91).

Comme en 2D, nous procédons par identification dans les trois dernières équations, ce qui

nous permet de déterminer le changement de variables sur les tenseurs des contraintes (3.90) :

$$\begin{cases} \sigma_{xx} &= -\sqrt{\rho}V_p(\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta))p^* \\ \sigma_{yy} &= -\sqrt{\rho}V_p(\kappa \cos^2 \theta + \sin^2 \theta(\kappa \cos^2 \phi + \sin^2 \phi))p^* \\ \sigma_{zz} &= -\sqrt{\rho}V_p(\kappa \sin^2 \theta + \cos^2 \theta)p^* \\ \sigma_{yz} &= \sqrt{\rho}V_p(\kappa - 1) \cos \theta \sin \theta \sin \phi p^* \\ \sigma_{xz} &= \sqrt{\rho}V_p(\kappa - 1) \cos \theta \sin \theta \cos \phi p^* \\ \sigma_{xy} &= \sqrt{\rho}V_p(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta p^* \end{cases} \quad (3.93)$$

Étape 4 : formulation de la CLA TTI elliptique pour les ondes P. Il suffit d'utiliser la CLA isotrope pour les ondes P : $p^* = -\rho V_p v_x^*$. La CLA TTI va s'écrire à partir de la CLA isotrope de la façon suivante. Le terme p^* est remplacé dans (3.93) pour exprimer σ_{xx} , σ_{xy} et σ_{xz} , qui sont les trois termes résultant du produit $\underline{\underline{\sigma}}\mathbf{n}$ lorsque $\mathbf{n} = \mathbf{e}_x$, en fonction de v_x^* . Puis, le terme v_x^* est exprimé en fonction de v_x , v_y et v_z grâce à (3.89). Ce processus conduit à la CLA TTI pour les ondes P :

$$\left\{ \begin{array}{l} \sigma_{xx} = -\rho V_p \frac{\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta)}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad + (\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta))v_x \\ \quad - (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad - (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] \\ \\ \sigma_{xz} = -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta \cos \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta))v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] \\ \\ \sigma_{xy} = -\rho V_p \frac{(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi(\kappa \cos^2 \theta + \sin^2 \theta))v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] \end{array} \right. \quad (3.94)$$

En résumé :

Exactement comme en 2D, en se plaçant dans le domaine de Fourier en espace, nous exprimons le champ anisotrope (TTI elliptique) en fonction du champ isotrope. Pour cela, (a) nous initialisons le procédé en exprimant les vecteurs d'ondes isotropes. L'initialisation

peut se faire car nous connaissons l'expression des surfaces de lenteur et donc les vecteurs d'ondes; **(b)** une fois les vecteurs d'ondes connus, nous pouvons exprimer le champ de vitesse après transformée de Fourier en espace – cette étape est facilitée par le fait qu'il est possible d'éliminer $\underline{\sigma}$ grâce à la formulation de l'équation à l'ordre deux; **(c)** nous connaissons le vecteur d'onde anisotrope et le champ de vitesse anisotrope, nous pouvons en déduire le tenseur $\underline{\underline{\sigma}}$; **(d)** les champs anisotropes s'exprimant en fonction des champs isotropes, nous appliquons la CLA isotrope et nous revenons aux champs anisotropes grâce au changement de variables inverse, ce qui nous permet finalement d'obtenir une CLA anisotrope (TTI elliptique).

CLA pour le système de l'élastodynamique TTI elliptique (avec $\gamma = 0$)

Comme en 2D, pour former une CLA TTI pour le système de l'élastodynamique, c'est-à-dire sans découpler les ondes P et les ondes S, il reste à additionner les contributions des CLA pour les ondes P elliptique et pour les ondes S. Nous considérons ici $\gamma = 0$ ce qui signifie que la CLA pour les ondes S est la même qu'en isotrope. Cette somme donne :

$$\left\{ \begin{array}{l} \sigma_{xx} = -\rho V_p \frac{\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad + (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad - (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad - (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] \\ \\ \sigma_{xz} = -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta \cos \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] - \rho V_s v_z \\ \\ \sigma_{xy} = -\rho V_p \frac{(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] - \rho V_s v_y \end{array} \right. \quad (3.95)$$

Remarque 3.8 Lorsque l'angle d'inclinaison θ est nul, la CLA TTI (3.41) fournit une CLA VTI car la rotation d'angle ϕ se fait alors selon l'axe de révolution de l'ellipsoïde aplati.

Remarque 3.9 Lorsque les constantes de Thomsen sont nulles (ie $\varepsilon = \delta = 0$, soit $\kappa = 1$), la CLA TTI (3.95) redevient isotrope, quels que soient les angles d'inclinaison. Ceci est dû au fait qu'en isotrope, les vitesses de propagation sont les mêmes dans toutes les directions.

3.3.2 Étude expérimentale préliminaire pour l'élastique TTI 3D

L'étude numérique en 2D a été réalisée sur des cas tests homogène et hétérogène avec des sources élastique et pseudo-acoustique. Les expériences en 3D étant plus longues à s'exécuter, nous n'avons pas encore tous les résultats. Nous présentons ici l'expérience sur un cas test homogène avec une source pseudo-acoustique en onde P, c'est-à-dire la configuration dans laquelle l'étude des réflexions est la plus intéressante.

Cas test

Nous considérons un cube homogène (voir Fig. 3.20) de dimension 10 km³ dont les caractéristiques physiques sont décrites dans le tableau 3.3. C'est une configuration d'un milieu elliptique simplifié et avec $\gamma = 0$.

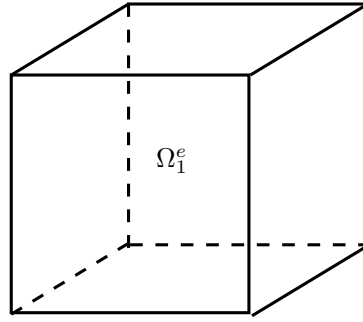


FIGURE 3.20 – Cas test TTI 3D homogène

milieu	ρ [kg.m ⁻³]	V_p [m.s ⁻¹]	V_s [m.s ⁻¹]	ε	δ	γ	θ [°]	ϕ [°]
Ω_1^e	1	4000	2000	0.25	0.25	0.00	30	15

TABLE 3.3 – Caractéristiques physiques du cas test TTI 3D

Source et condition initiale

De même que pour les cas 2D, nous allons suivre la procédure décrite dans la section 1.1.2 pour construire une condition initiale ou une source afin de ne générer qu'une onde P. Nous allons d'abord traiter le cas d'un milieu VTI puis nous en déduirons le cas d'une configuration TTI.

Dans un milieu VTI 3D, la matrice de l'équation de dispersion s'écrit :

$$E_{VTI,3D} = \begin{pmatrix} C_{11}k_x^2 + C_{66}k_y^2 + C_{44}k_z^2 & (C_{11} - C_{66})k_xk_y & (C_{44} + C_{13})k_xk_z \\ (C_{11} - C_{66})k_xk_y & C_{66}k_x^2 + C_{11}k_y^2 + C_{44}k_z^2 & (C_{44} + C_{13})k_yk_z \\ (C_{44} + C_{13})k_xk_z & (C_{44} + C_{13})k_yk_z & C_{44}k_x^2 + C_{44}k_y^2 + C_{33}k_z^2 \end{pmatrix} \quad (3.96)$$

où C est la matrice d'élasticité VTI 3D (1.20).

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

Dans le cas elliptique et lorsque $\gamma = 0$, les vecteurs propres de (3.96) sont orthogonaux et font intervenir les symboles de Fourier :

$$\Lambda_p = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} i k_x \\ \sqrt{\kappa^2 V_p^2 - V_s^2} i k_y \\ \sqrt{V_p^2 - V_s^2} i k_z \end{pmatrix}, \Lambda_{s_1} = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} k_x k_z \\ \sqrt{\kappa^2 V_p^2 - V_s^2} \sqrt{V_p^2 - V_s^2} k_y k_z \\ -(V_p^2 - V_s^2)(k_x^2 + k_y^2) \end{pmatrix}, \Lambda_{s_2} = \begin{pmatrix} -\sqrt{\kappa^2 V_p^2 - V_s^2} i k_y \\ \sqrt{\kappa^2 V_p^2 - V_s^2} i k_x \\ 0 \end{pmatrix} \quad (3.97)$$

ce qui conduit à l'opérateur pour l'onde P :

$$B_p = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_x \\ \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_y \\ \sqrt{V_p^2 - V_s^2} \partial_z \end{pmatrix} \quad (3.98)$$

Nous rappelons que nous avons choisi pour conditions initiales : $\underline{\sigma}_0 = 0$ et $\mathbf{v}_0(\mathbf{x}) = B_\star(e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2})$. Pour l'équation élastique VTI elliptique 3D (et $\gamma = 0$), cela conduit à :

$$\mathbf{v}_0(\mathbf{x}) = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} (x - x_0) \\ \sqrt{\kappa^2 V_p^2 - V_s^2} (y - y_0) \\ \sqrt{V_p^2 - V_s^2} (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \quad (3.99)$$

pour ne générer qu'une onde P.

Pour utiliser une source ponctuelle à la place d'une condition initiale, nous rappelons que sa forme, lorsqu'elle est appliquée sur l'équation du tenseur des contraintes, est $s_R(t) \underline{s}(\mathbf{x})$, où $s_R(t)$ est une amplitude (en l'occurrence une ondelette de Ricker définie en (1.9)). Il suffit ensuite de chercher \underline{s} tel que $\nabla \cdot \underline{s} = B_\star(\delta_{\mathbf{x}_0})$. Cette égalité donne pour B_p :

$$\begin{pmatrix} \partial_x s_{xx} + \partial_y s_{xy} + \partial_z s_{xz} \\ \partial_x s_{yx} + \partial_y s_{yy} + \partial_z s_{yz} \\ \partial_x s_{zx} + \partial_y s_{zy} + \partial_z s_{zz} \end{pmatrix} = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_x \\ \sqrt{\kappa^2 V_p^2 - V_s^2} \partial_y \\ \sqrt{V_p^2 - V_s^2} \partial_z \end{pmatrix} \delta_{\mathbf{x}_0} \quad (3.100)$$

L'identification terme à terme conduit à une solution simple : $s_{xz} = s_{zx} = s_{xy} = s_{yx} = s_{zy} = s_{yz} = 0$, $s_{xx} = s_{yy} = \sqrt{\kappa^2 V_p^2 - V_s^2} \delta_{\mathbf{x}_0}$ et $s_{zz} = \sqrt{V_p^2 - V_s^2} \delta_{\mathbf{x}_0}$. La source permettant

de ne générer qu'une onde P pour l'équation élastique VTI elliptique 3D (et $\gamma = 0$) s'écrit finalement :

$$\underline{\underline{s}}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) M_{VTI,P} \quad (3.101)$$

avec

$$M_{VTI,P} = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 & 0 \\ 0 & \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \\ 0 & 0 & \sqrt{V_p^2 - V_s^2} \end{pmatrix}$$

Comme nous l'avons remarqué en 2D (voir remarque 3.3), les sources et les conditions initiales TTI peuvent se déduire de l'étude VTI en utilisant la matrice $R_{(\theta,\phi)}$ du changement de base entre le repère cartésien et le repère TTI 3D, définie à l'équation (1.24). Ainsi, la source permettant de ne générer qu'une onde P pour l'équation élastique TTI elliptique 3D (et $\gamma = 0$) est $s_R(t)\underline{\underline{s}}(\mathbf{x})$ avec

$$\underline{\underline{s}}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) R_{(\theta,\phi)}^T M_{VTI,P} R_{(\theta,\phi)} \quad (3.102)$$

De même, la condition initiale permettant de ne générer qu'une onde P pour l'équation élastique TTI elliptique 3D (et $\gamma = 0$) est

$$\begin{cases} \underline{\underline{\sigma}}_0(\mathbf{x}) = 0 \\ \mathbf{v}_0(\mathbf{x}) = R_{(\theta,\phi)}^T M_{VTI,P} R_{(\theta,\phi)} \begin{pmatrix} (x - x_0) \\ (y - y_0) \\ (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2} \end{cases} \quad (3.103)$$

Résultats numériques

Les résultats sont présentés pour le modèle homogène avec une source en onde P. La Fig. 3.21 est une vue 3D tandis que les Fig. 3.24, 3.22 et 3.23 sont des vues 2D dans les différents plans du repère cartésien. Le module du vecteur vitesse est représenté à différents instants de simulation, choisis afin que les résultats soient comparables à l'étude 2D. À $t = 0.81s$, nous observons le front d'onde, ce qui permet de figer l'échelle des représentations. À $t = 1.32s$ et $t = 2.17s$, nous commençons à observer le fonctionnement des CLA car le front d'onde a atteint les bords du domaine de calcul. Nous laissons ensuite les calculs continuer jusqu'à $t = 3.26s$. Nous observons ainsi les réflexions parasites propres aux bords absorbants. Les mêmes résultats sont ensuite présentés à l'échelle de couleurs des réflexions, déterminée au temps $t = 3.26s$.

Comparaison avec la CLA isotrope

Nous nous proposons maintenant de comparer la CLA TTI à la CLA isotrope dans le domaine homogène TTI elliptique avec une source pseudo-acoustique en onde P. Les résultats sont représentés en 3D pour la CLA isotrope Fig. 3.25 (à comparer avec la Fig. 3.21). Les résultats sont également présentés sur des vues 2D des différents plans du repère cartésien sur les Fig. 3.28, 3.26 et 3.27. Comme pour l'étude 2D, il apparaît avant toute chose que les réflexions sont assez faibles même en utilisant une CLA isotrope. La différence la plus

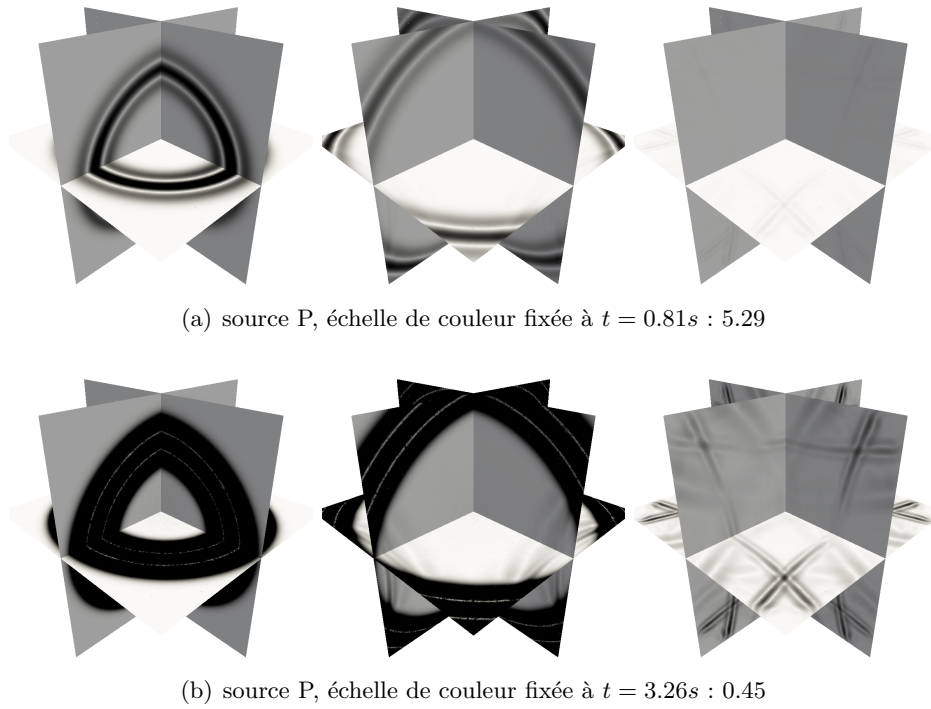
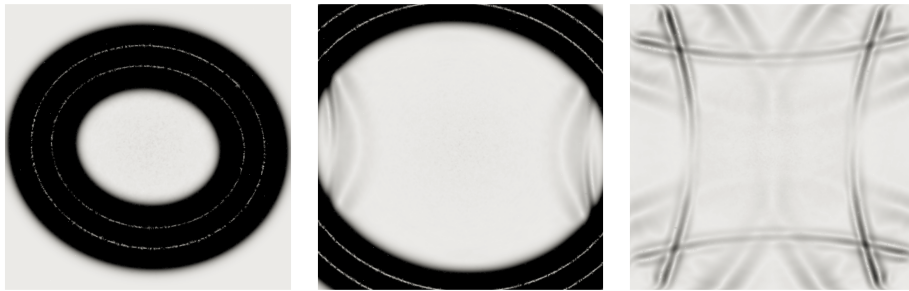


FIGURE 3.21 – Module du vecteur vitesse au cours du temps à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 3D

importante réside dans la réflexion de l'onde P en onde P qui n'apparaît pas dans le cas de la CLA TTI et qui est très visible dans le cas de la CLA isotrope. Ces réflexions sont plus fortes dans le plan (x, y) que pour un bord de normale extérieure \mathbf{e}_z . Cela vient du fait que les angles TTI sont assez faibles et donc la propagation des ondes dans l'axe vertical est proche d'un comportement isotrope. Le phénomène de réflexion d'onde P en onde S est aussi plus fort avec une CLA isotrope : ceci peut notamment se vérifier en comparant les échelles de représentation des calculs. Enfin, les réflexions de l'onde S sont du même ordre dans les deux cas et cela est cohérent puisque l'onde S est à chaque fois modélisée de la même façon.



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.29$



(b) source P, échelle de couleur fixée à $t = 3.26s : 0.45$

FIGURE 3.22 – Module du vecteur vitesse au cours du temps à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (y, z)



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.29$

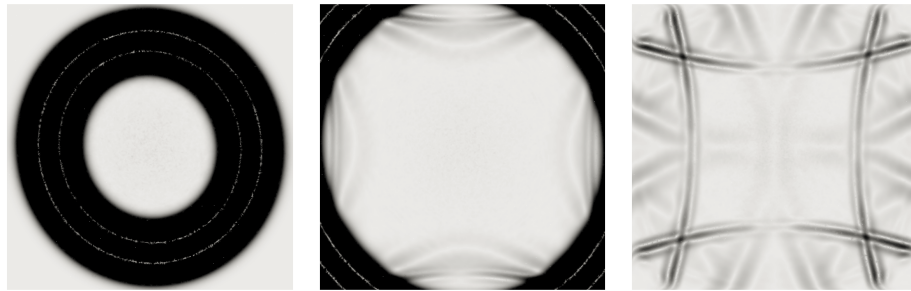


(b) source P, échelle de couleur fixée à $t = 3.26s : 0.45$

FIGURE 3.23 – Module du vecteur vitesse au cours du temps à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (x, z)

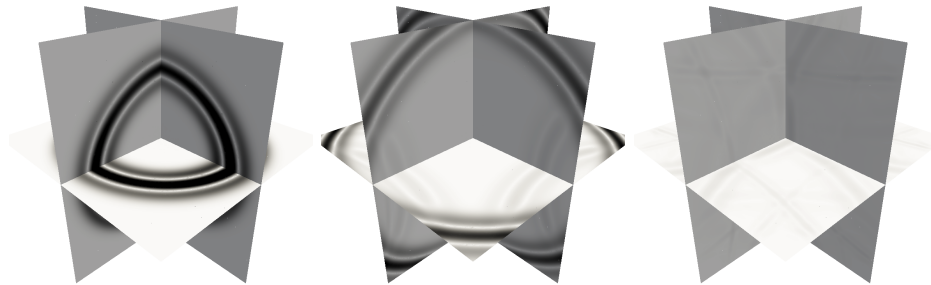


(a) source P, échelle de couleur fixée à $t = 0.81s : 5.29$

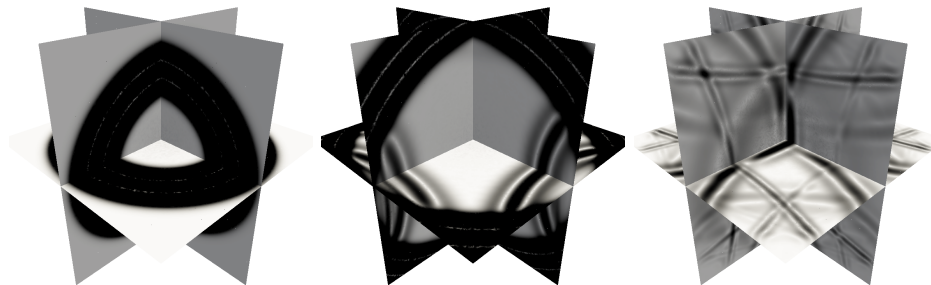


(b) source P, échelle de couleur fixée à $t = 3.26s : 0.45$

FIGURE 3.24 – Module du vecteur vitesse au cours du temps à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (x, y)



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.32$

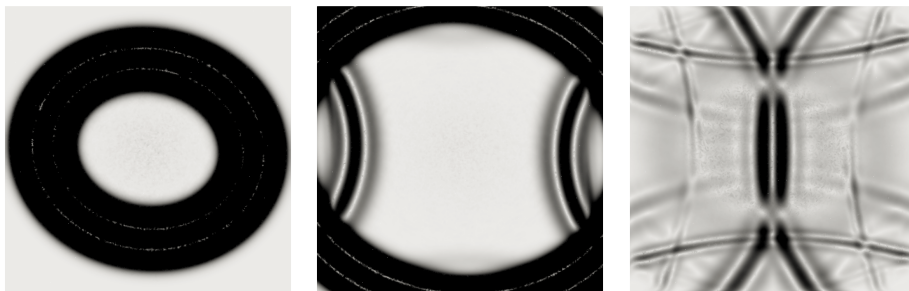


(b) source P, échelle de couleur fixée à $t = 3.26s : 0.38$

FIGURE 3.25 – Comparaison avec la CLA isotrope à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 3D



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.32$



(b) source P, échelle de couleur fixée à $t = 3.26s : 0.38$

FIGURE 3.26 – Comparaison avec la CLA isotrope à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (y, z)



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.32$

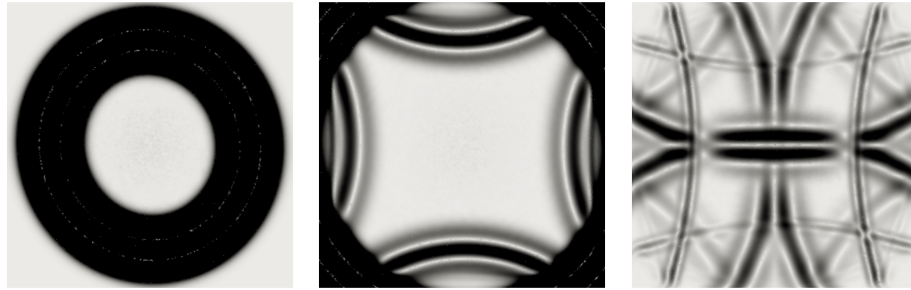


(b) source P, échelle de couleur fixée à $t = 3.26s : 0.38$

FIGURE 3.27 – Comparaison avec la CLA isotrope à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (x, z)



(a) source P, échelle de couleur fixée à $t = 0.81s : 5.32$



(b) source P, échelle de couleur fixée à $t = 3.26s : 0.38$

FIGURE 3.28 – Comparaison avec la CLA isotrope à $t = 0.81s$, $1.32s$ et $3.26s$ (de gauche à droite) dans la configuration homogène TTI 3D, vue 2D dans le plan (x, y)

3.3.3 Stabilité des problèmes mixtes TTI 3D

Comme à la sous-section 3.1.4, le système de l'élastodynamique (1.5) est lié à la fonction $\mathbb{E}(t) = \frac{1}{2}(\|v\|_{\Omega_x, \rho}^2 + \|\underline{\sigma}\|_{\Omega_x, \underline{C}^{-1}}^2)$. Nous supposons que la solution du problème est assez régulière pour avoir :

$$\frac{d}{dt}\mathbb{E}(t) = \int_{\Gamma_x} (\underline{\sigma}\mathbf{n})v d\mathbf{x} \quad (3.104)$$

Nous nous plaçons sur une frontière $\Gamma_x = \{x = 0\}$. Le vecteur \mathbf{n} est la normale extérieure à Γ_x donc $\mathbf{n} = (1, 0, 0)$ et $\underline{\sigma}\mathbf{n} = (\sigma_{xx}, \sigma_{yy}, \sigma_{xz})$. Enfin, les composantes de $\underline{\sigma}$ sont définies par la CLA TTI du système (3.95). Ainsi :

$$\left\{ \begin{aligned} \frac{d}{dt}\mathbb{E}(t) &= - \int_{\Gamma_x} \left[\frac{\rho V_p}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} \right. \\ &\quad \left[(\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \right. \\ &\quad \left. - (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y - (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z \right]^2 \\ &\quad \left. + \rho V_s (v_y^2 + v_z^2) \right] d\mathbf{x} \end{aligned} \right. \quad (3.105)$$

Comme cette quantité est négative, l'énergie du système continu est décroissante, ce qui assure la stabilité du problème de l'élastodynamique 3D TTI avec la CLA (3.95).

En suivant les mêmes étapes qu'au chapitre précédent, nous pouvons affirmer que l'énergie

$$\mathbb{E}_{semi_d}(t) = \langle \partial_t M_{\mathbf{v}} \mathbf{v}_h, \mathbf{v}_h \rangle + \langle \partial_t M_{\sigma, C^{-1}} \sigma_h, \sigma_h \rangle \quad (3.106)$$

associée au problème mixte semi-discret obtenu en appliquant une formulation DG avec flux centrés est décroissante. Enfin, pour que la quantité

$$\mathbb{E}_d^n = \frac{1}{2} \langle M_{\mathbf{v}} \mathbf{v}_h^n, \mathbf{v}_h^n \rangle + \frac{1}{2} \langle M_{\sigma, C^{-1}} \sigma_h^{n+\frac{1}{2}}, \sigma_h^{n-\frac{1}{2}} \rangle \quad (3.107)$$

associée au problème mixte discret obtenu en combinant au problème semi-discret un schéma en temps de type Leap-Frog soit une énergie, il faut et il suffit que la matrice

$$M_{\mathbf{v}} - \frac{\Delta t^2}{4} R_{\sigma} (M_{\sigma, C})^{-1} (R_{\sigma})^T \quad (3.108)$$

soit positive, ce qui définit une condition de stabilité du schéma de type CFL

$$1 - \frac{\Delta t^2}{4} \lambda_{max} \geq 0 \quad (3.109)$$

où λ_{max} est la plus grande valeur propre de la matrice $M_{\mathbf{v}}^{-1} R_{\sigma} (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T$.

Remarque 3.10 Dans le code DIVA, nous mesurons l'énergie discrète suivant la formule (3.107). Nous avons également modifié la CFL en implémentant la méthode de la puissance (voir algorithme 2.1), qui a l'avantage de s'appliquer indépendamment à des configurations isotropes et anisotropes.

3.3.4 Courbes d'énergies de l'étude préliminaire TTI 3D

En utilisant une CLA isotrope dans un milieu TTI, nous avons observé à la Fig. 3.25 que les réflexions d'ondes P étaient très fortes. L'étude de l'énergie sur le cas homogène de cette expérience est présentée sur la Fig. 3.29. Les expériences en 3D étant plus longues à réaliser, nous présentons ici seulement les résultats pour une source en onde P dans un milieu homogène TTI elliptique. Les courbes confirment les observations, les réflexions de l'onde P transformée en onde P et en onde S forment des paliers distincts. Elles sont plus fortes avec une CLA isotrope qu'avec une CLA TTI (exactement comme en 2D, voir Fig. 3.10).

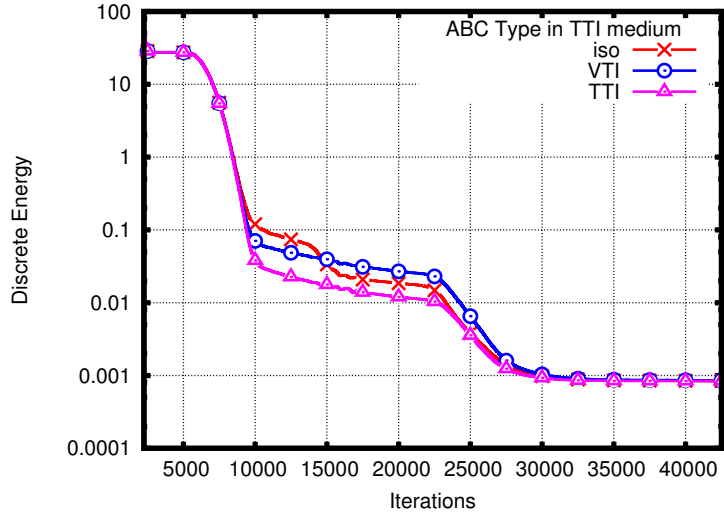


FIGURE 3.29 – Comparaison avec les CLA isotrope et VTI dans un milieu homogène TTI 3D avec une source P

3.3.5 Cas d'une frontière artificielle plane générale

Dans cette section, nous montrons comment étendre la CLA formulée précédemment pour une frontière plane parallèle à un des axes au cas d'une frontière plane de normale $\mathbf{n} = (n_x, n_y, n_z)$. Nous présentons uniquement la construction en 3D, le cas 2D pouvant être considéré comme un cas particulier avec $n_y = 0$ et $\phi = 0$.

Nous définissons tout d'abord le vecteur TTI définissant la direction d'anisotropie :

$$\mathbf{u} = \begin{pmatrix} \cos \phi \sin \theta \\ \sin \phi \sin \theta \\ \cos \theta \end{pmatrix} \quad (3.110)$$

Dans le cas $\mathbf{n} = \mathbf{e}_x$, nous avons vu à la section précédente que la CLA TTI s'écrivait sous la forme générique :

$$\begin{cases} \sigma_{xx} &= a_{xx}(\theta, \phi)v_x + a_{xy}(\theta, \phi)v_y + a_{xz}(\theta, \phi)v_z \\ \sigma_{xy} &= a_{yx}(\theta, \phi)v_x + a_{yy}(\theta, \phi)v_y + a_{yz}(\theta, \phi)v_z \\ \sigma_{xz} &= a_{zx}(\theta, \phi)v_x + a_{zy}(\theta, \phi)v_y + a_{zz}(\theta, \phi)v_z \end{cases} \quad (3.111)$$

Nous allons déterminer l'équation de la CLA en quatre étapes : **(a)** nous déterminons une base orthonormée $(\mathbf{n}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ et la matrice de rotation P transformant $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ en $(\mathbf{n}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ où $\boldsymbol{\tau}$ désigne le vecteur tangent à la surface artificielle ; **(b)** nous exprimons le vecteur TTI dans la nouvelle base à l'aide d'une matrice notée P ; **(c)** nous pouvons alors écrire l'équation de la CLA dans le repère $(\mathbf{n}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$; **(d)** nous terminons en écrivant l'équation de la CLA dans le repère $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ à l'aide de la matrice P .

Étape 1 : déterminer les vecteurs $\boldsymbol{\tau}_1$ et $\boldsymbol{\tau}_2$ pour former une base orthonormée avec \mathbf{n} . Il faut distinguer 2 cas :

- si $n_x^2 + n_y^2 \neq 0$, et donc $|n_z| \neq 1$
nous appliquons une rotation d'angle $\cos \psi_1 = \frac{n_x}{\sqrt{n_x^2 + n_y^2}}$, $\sin \psi_1 = \frac{n_y}{\sqrt{n_x^2 + n_y^2}}$ autour de l'axe \mathbf{e}_z du repère $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. Cette rotation est caractérisée par la matrice

$$P_1 = \begin{pmatrix} \frac{n_x}{\sqrt{n_x^2 + n_y^2}} & -\frac{n_y}{\sqrt{n_x^2 + n_y^2}} & 0 \\ \frac{n_y}{\sqrt{n_x^2 + n_y^2}} & \frac{n_x}{\sqrt{n_x^2 + n_y^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Nous obtenons un nouveau repère $(\mathbf{e}'_x, \mathbf{e}'_y, \mathbf{e}'_z)$ tel que

$$\mathbf{e}'_x = \begin{pmatrix} \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ \frac{n_y}{\sqrt{n_x^2 + n_y^2}} \\ 0 \end{pmatrix}, \mathbf{e}'_y = \begin{pmatrix} -\frac{n_y}{\sqrt{n_x^2 + n_y^2}} \\ \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ 0 \end{pmatrix}, \mathbf{e}'_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \mathbf{e}_z$$

Nous appliquons ensuite une rotation d'angle $\cos \psi_2 = \frac{n_z}{\sqrt{n_x^2 + n_y^2}}$, $\sin \psi_2 = \frac{n_z}{\sqrt{n_x^2 + n_y^2}}$ et d'axe $-\mathbf{e}'_y$ au repère $(\mathbf{e}'_x, \mathbf{e}'_y, \mathbf{e}'_z)$. Cette rotation est caractérisée par la matrice

$$P_2 = \begin{pmatrix} \frac{n_y^2}{n_x^2 + n_y^2} + \frac{n_x^2}{n_x^2 + n_y^2} \sqrt{n_x^2 + n_y^2} & -\frac{n_x n_y}{n_x^2 + n_y^2} (1 - \sqrt{n_x^2 + n_y^2}) & -\frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} \\ -\frac{n_x n_y}{n_x^2 + n_y^2} (1 - \sqrt{n_x^2 + n_y^2}) & \frac{n_x^2}{n_x^2 + n_y^2} + \frac{n_y^2}{n_x^2 + n_y^2} \sqrt{n_x^2 + n_y^2} & -\frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \\ \frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} & \frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} & \sqrt{n_x^2 + n_y^2} \end{pmatrix}$$

et la matrice de rotation s'écrit

$$P = P_1 P_2 = \begin{pmatrix} n_x & -\frac{n_y}{\sqrt{n_x^2 + n_y^2}} & -\frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} \\ n_y & \frac{n_x}{\sqrt{n_x^2 + n_y^2}} & -\frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \\ n_z & 0 & \sqrt{n_x^2 + n_y^2} \end{pmatrix}$$

Nous en déduisons donc le nouveau repère

$$\mathbf{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix}, \boldsymbol{\tau}_1 = \begin{pmatrix} -\frac{n_y}{\sqrt{n_x^2 + n_y^2}} \\ \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ 0 \end{pmatrix}, \boldsymbol{\tau}_2 = \begin{pmatrix} -\frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} \\ -\frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \\ \sqrt{n_x^2 + n_y^2} \end{pmatrix}$$

Chapitre 3. Conditions aux limites absorbantes pour des matériaux TTI

– si $n_x^2 + n_y^2 = 0$, i.e. $|n_z| = 1$.

Il suffit d'appliquer une rotation d'angle $n_z \frac{\pi}{2}$ autour de $-\mathbf{e}'_y$ pour obtenir le nouveau repère. Nous avons alors

$$P = \begin{pmatrix} \cos(n_z \frac{\pi}{2}) = 0 & 0 & -\sin(n_z \frac{\pi}{2}) \\ 0 & 1 & 0 \\ \sin(n_z \frac{\pi}{2}) & 0 & \cos(n_z \frac{\pi}{2}) = 0 \end{pmatrix}$$

soit

$$P = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ si } n_z = 1, \text{ et } P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \text{ si } n_z = -1$$

Nous avons alors

$$\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tau_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tau_2 = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \text{ si } n_z = 1$$

$$\mathbf{n} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \tau_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \tau_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \text{ si } n_z = -1$$

Étape 2 : Nous pouvons maintenant exprimer le vecteur de direction TTI dans le nouveau repère $\mathbf{u}' = P\mathbf{u}$. Nous obtenons

$$\mathbf{u}' = \begin{pmatrix} (n_x \cos \phi + n_y \sin \phi) \sin \theta + n_z \cos \theta \\ (\frac{-n_y}{\sqrt{n_x^2 + n_y^2}} \cos \phi + \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \sin \phi) \sin \theta \\ (\frac{-n_x n_z}{\sqrt{n_x^2 + n_y^2}} \cos \phi - \frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \sin \phi) \sin \theta + \sqrt{n_x^2 + n_y^2} \cos \theta \end{pmatrix} \text{ si } |n_z| \neq 1$$

$$\mathbf{u}' = \begin{pmatrix} \cos \theta \\ \sin \phi \sin \theta \\ -\cos \phi \sin \theta \end{pmatrix} \text{ si } n_z = 1, \text{ et } \mathbf{u}' = \begin{pmatrix} -\cos \theta \\ \sin \phi \sin \theta \\ \cos \phi \sin \theta \end{pmatrix} \text{ si } n_z = -1$$

Nous en déduisons alors les sinus et cosinus des angles θ' et ϕ' caractérisant le milieu TTI dans le repère $(\mathbf{n}, \tau_1, \tau_2)$ en utilisant le fait que

$$\mathbf{u}' = \begin{pmatrix} \cos \phi' \sin \theta' \\ \sin \phi' \sin \theta' \\ \cos \theta' \end{pmatrix}$$

Nous avons donc

$$\cos \theta' = \left(\frac{-n_x n_z}{\sqrt{n_x^2 + n_y^2}} \cos \phi - \frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \sin \phi \right) \sin \theta + \sqrt{n_x^2 + n_y^2} \cos \theta \text{ si } |n_z| \neq 1$$

et

$$\cos \theta' = -n_z \cos \phi \sin \theta \text{ si } |n_z| = 1$$

Nous posons $\sin \theta' = \sqrt{1 - \cos^2 \theta'}$ (par convention héritée des coordonnées sphériques, le sinus de l'angle θ est toujours positif). Il nous reste maintenant à déterminer $\cos \phi'$ et $\sin \phi'$. Nous avons

$$\begin{cases} \cos \phi' \sin \theta' &= (n_x \cos \phi + n_y \sin \phi) \sin \theta + n_z \cos \theta \\ \sin \phi' \sin \theta' &= \left(\frac{-n_y}{\sqrt{n_x^2 + n_y^2}} \cos \phi + \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \sin \phi \right) \sin \theta \end{cases}$$

si $|n_z| \neq 1$ et

$$\begin{cases} \cos \phi' \sin \theta' &= n_z \cos \theta \\ \sin \phi' \sin \theta' &= \sin \phi \sin \theta \end{cases}$$

si $|n_z| = 1$.

Il faut donc distinguer à nouveau 2 cas :

– si $\sin \theta' \neq 0$, alors

$$\begin{cases} \cos \phi' &= (n_x \cos \phi + n_y \sin \phi) \frac{\sin \theta}{\sin \theta'} + n_z \frac{\cos \theta}{\sin \theta'} \\ \sin \phi' &= \left(\frac{-n_y \cos \phi}{\sqrt{n_x^2 + n_y^2}} + \frac{n_x \sin \phi}{\sqrt{n_x^2 + n_y^2}} \right) \frac{\sin \theta}{\sin \theta'} \end{cases}$$

si $|n_z| \neq 1$ et

$$\begin{cases} \cos \phi' &= \frac{n_z \cos \theta}{\sin \theta'} \\ \sin \phi' &= \frac{\sin \phi \sin \theta}{\sin \theta'} \end{cases}$$

si $|n_z| = 1$.

– si $\sin \theta' = 0$, alors nous pouvons choisir $\cos \phi'$ et $\sin \phi'$ arbitrairement car l'anisotropie est dirigée suivant $\boldsymbol{\tau}_2$. Pour simplifier, nous prenons $\cos \phi' = 1$ et $\sin \phi' = 0$.

Étape 3 : nous pouvons maintenant exprimer la CLA dans le repère $(\mathbf{n}, \boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$

$$\begin{cases} \underline{\underline{\sigma}} \mathbf{n} \cdot \mathbf{n} &= a_{xx}(\theta', \phi') \mathbf{v} \cdot \mathbf{n} + a_{xy}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_1 + a_{xz}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_2 \\ \underline{\underline{\sigma}} \mathbf{n} \cdot \boldsymbol{\tau}_1 &= a_{yx}(\theta', \phi') \mathbf{v} \cdot \mathbf{n} + a_{yy}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_1 + a_{yz}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_2 \\ \underline{\underline{\sigma}} \mathbf{n} \cdot \boldsymbol{\tau}_2 &= a_{zx}(\theta', \phi') \mathbf{v} \cdot \mathbf{n} + a_{zy}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_1 + a_{zz}(\theta', \phi') \mathbf{v} \cdot \boldsymbol{\tau}_2 \end{cases} \quad (3.112)$$

Étape 4 : il reste maintenant à revenir dans le repère $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. Nous avons

$$\begin{cases} \mathbf{v} \cdot \mathbf{n} &= v_x n_x + v_y n_y + v_z n_z \\ \mathbf{v} \cdot \boldsymbol{\tau}_1 &= -v_x \frac{n_y}{\sqrt{n_x^2 + n_y^2}} + v_y \frac{n_x}{\sqrt{n_x^2 + n_y^2}} \\ \mathbf{v} \cdot \boldsymbol{\tau}_2 &= -v_x \frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} - v_y \frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} + v_z \sqrt{n_x^2 + n_y^2} \end{cases} \quad (3.113)$$

Nous pouvons donc écrire la CLA sous la forme

$$\begin{cases} \underline{\underline{\sigma}} \mathbf{n} \cdot \mathbf{n} &= \alpha_{nx} v_x + \alpha_{ny} v_y + \alpha_{nz} v_z \\ \underline{\underline{\sigma}} \mathbf{n} \cdot \boldsymbol{\tau}_1 &= \alpha_{\tau_1 x} v_x + \alpha_{\tau_1 y} v_y + \alpha_{\tau_1 z} v_z \\ \underline{\underline{\sigma}} \mathbf{n} \cdot \boldsymbol{\tau}_2 &= \alpha_{\tau_2 x} v_x + \alpha_{\tau_2 y} v_y + \alpha_{\tau_2 z} v_z \end{cases} \quad (3.114)$$

Il ne reste plus qu'à exprimer les vecteurs $\underline{\underline{\sigma}}\mathbf{n}$ dans la base $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. Nous utilisons pour cela la matrice P :

$$\begin{pmatrix} \underline{\underline{\sigma}}\mathbf{n} \cdot \mathbf{e}_x \\ \underline{\underline{\sigma}}\mathbf{n} \cdot \mathbf{e}_y \\ \underline{\underline{\sigma}}\mathbf{n} \cdot \mathbf{e}_z \end{pmatrix} = P \begin{pmatrix} \alpha_{nx}v_x + \alpha_{ny}v_y + \alpha_{nz}v_z \\ \alpha_{\tau_1 x}v_x + \alpha_{\tau_1 y}v_y + \alpha_{\tau_1 z}v_z \\ \alpha_{\tau_2 x}v_x + \alpha_{\tau_2 y}v_y + \alpha_{\tau_2 z}v_z \end{pmatrix}$$

Nous pouvons également tout exprimer sous forme matricielle. (3.112) se réécrit $(\underline{\underline{\sigma}}\mathbf{n})_{\mathbf{n}, \tau_1, \tau_2} = A\mathbf{v}_{\mathbf{n}, \tau_1, \tau_2}$. Nous avons $\mathbf{v}_{\mathbf{n}, \tau_1, \tau_2} = P^{-1}\mathbf{v}_{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z} = P^T\mathbf{v}_{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z}$ et $(\underline{\underline{\sigma}}\mathbf{n})_{\mathbf{n}, \tau_1, \tau_2} = P^{-1}(\underline{\underline{\sigma}}\mathbf{n})_{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z}$. Donc $\boxed{(\underline{\underline{\sigma}}\mathbf{n})_{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z} = PAP^T\mathbf{v}_{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z}}$.

3.4 Conclusion

L'approche proposée par Engquist et Majda s'avère très compliquée à mettre en œuvre dans le cas d'un milieu TTI et ceci même si nous nous limitons au cas 2D. Cela peut s'expliquer par le fait que cette technique de construction s'appuie sur la représentation algébrique du système d'équations aux dérivées partielles *via* le calcul symbolique. Or, le passage au système algébrique ne simplifie en rien les paramètres caractérisant le milieu. Notamment, les paramètres gouvernant l'inclinaison de l'axe d'orthotropie contribuent au remplissage du tenseur des contraintes. Nous sommes donc face au calcul des valeurs propres d'un tenseur plein à coefficients variables dont les formules représentatives sont difficilement exploitables. Nous avons donc proposé une autre approche pour le cas d'un milieu TTI qui est basée sur le cas isotrope. Cette démarche permet de contourner les difficultés que nous venons d'exposer et est rendue possible car l'on peut lier les caractérisations des milieux isotrope et TTI par des changements de variables. Nous pouvons ainsi transformer le système TTI elliptique en un système isotrope pour lequel nous connaissons l'expression d'une CLA et en déduire une CLA adaptée au milieu anisotrope en appliquant le changement de variables inverse.

Comme nous avons remarqué qu'une CLA VTI d'ordre faible ne fait pas intervenir le coefficient δ , nous avons décidé de considérer un cas d'anisotropie elliptique (i.e. $\delta = \varepsilon$). Cette configuration est très intéressante car dans ce cas, les courbes de lenteur associées au système sont facilement paramétrables, à savoir elles décrivent une ellipse inclinée pour des ondes P et un cercle pour des ondes S. Un autre point important est le fait que dans un milieu isotrope ou VTI, la CLA d'ordre faible était égale à la somme des contributions des CLA des sous-problèmes pseudo-acoustiques. Nous avons alors opté pour le découplage des problèmes pour les ondes P et les ondes S, ce qui simplifie beaucoup la méthode de construction. Nous pouvons en effet nous limiter à la construction de CLA pour les ondes P car en étudiant les courbes de lenteur associées aux ondes S, nous observons que la CLA d'ordre faible TTI elliptique 2D pour les ondes S est la même que pour un milieu isotrope. La construction de la CLA d'ordre faible TTI elliptique pour les ondes P se fait, quant à elle, en plusieurs étapes. Le changement de variables sur les formes géométriques des courbes de lenteur TTI elliptique et isotrope traduit une relation sur les vecteurs d'ondes. En comparant ensuite les formulations au deuxième ordre des systèmes de l'élastodynamique TTI elliptique et isotrope dans le domaine de Fourier, il vient un changement de variables sur les vecteurs vitesse. Puis, les formulations au premier ordre permettent de déduire une relation sur les tenseurs de contraintes. Il ne reste plus qu'à injecter dans la CLA isotrope pour les ondes P les changements de variables sur les vitesses et les tenseurs des contraintes pour obtenir une CLA TTI elliptique 2D pour les ondes P.

Nous avons ensuite étendu le processus de construction à la dimension trois d'espace. La CLA TTI elliptique 3D pour les ondes P s'obtient en suivant les mêmes étapes, la surface de lenteur associée formant un ellipsoïde incliné. Le cas des ondes S est plus délicat car elles se dédoublent en 3D et l'une d'elle est déformée par le coefficient TI γ . Nous avons considéré ici $\gamma = 0$, ce qui permet de définir la CLA d'ordre faible TTI elliptique 3D pour les ondes S comme celle de l'isotrope. Il s'agit d'un résultat préliminaire somme toute intéressant car il s'avère que les géophysiciens se concentrent principalement sur les ondes P. L'étude du cas $\gamma \neq 0$ est bien évidemment à faire et constitue une des perspectives de ce travail. Enfin, nous avons proposé une généralisation dans le cas d'une frontière plane quelconque, c'est-à-dire non parallèle à un des axes du repère. Cette généralisation n'est pas un point clé de notre travail car en pratique les calculs numériques sont réalisés dans des boîtes dont les côtés sont parallèles à un des axes du repère. Toutefois, il pourrait s'avérer intéressant d'utiliser des frontières planes "inclinaées" pour réduire la taille du système.

Afin de mieux évaluer les réflexions dues aux CLA proposées, nous avons construit des sources et des conditions initiales permettant de ne générer qu'une onde P ou qu'une onde S dans un matériau TTI elliptique 2D et qu'une onde P dans un matériau TTI elliptique 3D. Nous avons été étonnés de ne rien trouver à ce sujet dans la littérature car la construction de données est un point clé dans l'élaboration de cas test. L'étude expérimentale que nous avons faite à 2D dans un milieu TTI a révélé un comportement similaire aux précédentes études dans un milieu isotrope ou VTI (elliptique ou non), tant sur la localisation que sur l'intensité des réflexions parasites. En comparant ensuite la CLA TTI et les CLA isotrope et VTI cette fois dans un même milieu TTI 2D (elliptique ou non), nous avons illustré la supériorité de la CLA TTI, essentiellement sur la composante en onde P, la composante en onde S étant strictement la même dans tous les cas. L'étude expérimentale 3D, bien que préliminaire, montre des réflexions parasites cohérentes avec ce qui est observé à 2D, ainsi que l'intérêt d'utiliser la CLA TTI plutôt que la CLA isotrope dans un milieu TTI 3D.

Nous avons aussi étudié la stabilité du problème mixte continu résultant du couplage de l'élastodynamique avec la CLA TTI 2D ou 3D et du problème semi-discret associé écrit en formulation DG avec flux centrés. De la même manière que pour le cas VTI, la stabilité des énergies associées aux problèmes continu et semi-discret (formulation DG à flux centrés) sont obtenues formellement grâce à la positivité des paramètres physiques intervenant dans les équations. Le problème discret (schéma en temps de type Leap-Frog) permet de définir une énergie discrète, c'est-à-dire une fonctionnelle positive à chaque pas de temps, sous une condition de type CFL. La stabilité du problème discret découle alors de la décroissance de l'énergie discrète. Les résultats de stabilité sont illustrés par des courbes d'énergies qui montrent aussi que la CLA TTI 2D ou 3D fonctionne mieux que les CLA isotrope ou VTI dans un milieu TTI 2D ou 3D (elliptique ou non).

Pour finir, nous avons tracé les coefficients de réflexion A_{PP} , A_{PS} , A_{SP} et A_{SS} de la CLA TTI 2D pour un milieu TTI elliptique fixé, en fonction de l'angle d'incidence θ_0 pour l'onde S et du paramètre k_z du vecteur d'onde pour l'onde P. Dans le cas TTI, les coefficients ne sont plus symétriques et ils ne sont plus nuls à incidence normale (i.e. lorsque $\theta_0 = 0$). Néanmoins, leurs comportements présentent des similitudes avec les courbes VTI : il existe au moins un point, hors extrémités, sur chaque courbe de CLA TTI où les réflexions sont nulles. De plus, nous avons montré que la CLA TTI 2D était meilleure que les CLA isotrope ou VTI dans un milieu TTI elliptique. L'étude des coefficients de réflexion de la CLA TTI 3D est une autre de nos perspectives de travail.

En résumé, nous avons obtenu les résultats suivants :

Théorème 3.1 Une CLA pour des matériaux TTI 2D dans le cas d'une frontière plane de normale extérieure \mathbf{e}_x est :

$$\begin{cases} \sigma_{xx} &= -\rho V_p \frac{\kappa \cos^2 \theta + \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [(\kappa \cos^2 \theta + \sin^2 \theta) v_x - (\kappa - 1) \cos \theta \sin \theta v_z] \\ \sigma_{xz} &= -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta}{\sqrt{\kappa^2 \cos^2 \theta + \sin^2 \theta}} [-(\kappa \cos^2 \theta + \sin^2 \theta) v_x + (\kappa - 1) \cos \theta \sin \theta v_z] - \rho V_s v_z \end{cases} \quad (3.115)$$

et une CLA pour des matériaux TTI 3D dans le cas d'une frontière plane de normale extérieure \mathbf{e}_x est :

$$\begin{cases} \sigma_{xx} = -\rho V_p \frac{\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad + (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad - (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad - (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] \\ \sigma_{xz} = -\rho V_p \frac{(\kappa - 1) \cos \theta \sin \theta \cos \phi}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] - \rho V_s v_z \\ \sigma_{xy} = -\rho V_p \frac{(\kappa - 1) \cos \phi \sin \phi \sin^2 \theta}{\sqrt{\kappa^2 \cos^2 \theta \cos^2 \phi + \kappa^2 \sin^2 \phi + \sin^2 \theta \cos^2 \phi}} [\\ \quad - (\kappa \sin^2 \phi + \cos^2 \phi (\kappa \cos^2 \theta + \sin^2 \theta)) v_x \\ \quad + (\kappa - 1) \cos \phi \sin \phi \sin^2 \theta v_y \\ \quad + (\kappa - 1) \cos \theta \sin \theta \cos \phi v_z] - \rho V_s v_y \end{cases} \quad (3.116)$$

Théorème 3.2 La CLA TTI 3D peut être formulée pour une frontière plane de normale extérieure quelconque $\mathbf{n} = (n_x, n_y, n_z)$. Soit A la matrice formée des coefficients de la CLA TTI 3D (3.116) obtenue dans le cas d'une frontière plane de normale extérieure \mathbf{e}_x telle que $\underline{\underline{\sigma}} \mathbf{e}_x = A(\theta, \phi) \mathbf{v}$. Soit θ' défini par :

$$\begin{cases} \cos \theta' &= \left(\frac{-n_x n_z}{\sqrt{n_x^2 + n_y^2}} \cos \phi - \frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \sin \phi \right) \sin \theta + \sqrt{n_x^2 + n_y^2} \cos \theta \text{ si } |n_z| \neq 1 \\ \cos \theta' &= -n_z \cos \phi \sin \theta \text{ si } |n_z| = 1 \end{cases}$$

et

$$\sin \theta' = \sqrt{1 - \cos^2 \theta'}.$$

Soit ϕ' défini par :

– si $\sin \theta' \neq 0$,

$$\begin{cases} \cos \phi' &= (n_x \cos \phi + n_y \sin \phi) \frac{\sin \theta}{\sin \theta'} + n_z \frac{\cos \theta}{\sin \theta'} \\ \sin \phi' &= \left(\frac{-n_y \cos \phi}{\sqrt{n_x^2 + n_y^2}} + \frac{n_x \sin \phi}{\sqrt{n_x^2 + n_y^2}} \right) \frac{\sin \theta}{\sin \theta'} \end{cases}, \quad \text{si } |n_z| \neq 1$$

$$\begin{cases} \cos \phi' &= \frac{n_z \cos \theta}{\sin \theta'} \\ \sin \phi' &= \frac{\sin \phi \sin \theta}{\sin \theta'} \end{cases}, \quad \text{si } |n_z| = 1$$

– si $\sin \theta' = 0$

$$\cos \phi' = 1 \text{ et } \sin \phi' = 0.$$

Soit la matrice P définie par

– si $|n_z| \neq 1$

$$\begin{pmatrix} n_x & -\frac{n_y}{\sqrt{n_x^2 + n_y^2}} & -\frac{n_x n_z}{\sqrt{n_x^2 + n_y^2}} \\ n_y & \frac{n_x}{\sqrt{n_x^2 + n_y^2}} & -\frac{n_y n_z}{\sqrt{n_x^2 + n_y^2}} \\ n_z & 0 & \sqrt{n_x^2 + n_y^2} \end{pmatrix}$$

– si $|n_z| = 1$

$$P = \begin{pmatrix} \cos(n_z \frac{\pi}{2}) = 0 & 0 & -\sin(n_z \frac{\pi}{2}) \\ 0 & 1 & 0 \\ \sin(n_z \frac{\pi}{2}) & 0 & \cos(n_z \frac{\pi}{2}) = 0 \end{pmatrix}$$

Alors la CLA TTI 3D pour une frontière plane de normale extérieure $\mathbf{n} = (n_x, n_y, n_z)$ est

$$\underline{\underline{\sigma}} \mathbf{n} = P A(\theta', \phi') P^T \mathbf{v}.$$

Théorème 3.3 Pour ne générer que des ondes P ou des ondes S dans un matériau TTI elliptique 2D, il suffit d'appliquer au système de l'élastodynamique non homogène :

$$\begin{cases} \rho(\mathbf{x}) \partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) + s_R(t) \underline{\underline{s}}(\mathbf{x}) \end{cases} \quad (3.117)$$

les sources :

$$\begin{cases} \underline{\underline{s}}(\mathbf{x}) &= \delta_{x_0}(\mathbf{x}) \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta) & (X - Z) \cos \theta \sin \theta \\ (X - Z) \cos \theta \sin \theta & (Z \cos^2 \theta + X \sin^2 \theta) \end{pmatrix}, \text{ pour ne générer qu'une onde } P, \\ \underline{\underline{s}}(\mathbf{x}) &= \delta_{x_0}(\mathbf{x}) \begin{pmatrix} -(X - Z) \cos \theta \sin \theta & -(Z \cos^2 \theta + X \sin^2 \theta) \\ (X \cos^2 \theta + Z \sin^2 \theta) & (X - Z) \cos \theta \sin \theta \end{pmatrix}, \text{ pour ne générer qu'une onde } S, \end{cases} \quad (3.118)$$

où $X = \sqrt{\kappa^2 V_p^2 - V_s^2}$ et $Z = \sqrt{V_p^2 - V_s^2}$ et, par exemple, s_R est donnée comme une ondelette de Ricker de fréquence pic f_p :

$$s_R(t) = (1 - 2\pi^2 f_p^2 t^2) e^{-\pi^2 f_p^2 t^2} \quad (3.119)$$

ou de considérer le problème de Cauchy :

$$\begin{cases} \rho(\mathbf{x})\partial_t \mathbf{v}(\mathbf{x}, t) &= \nabla \cdot \underline{\underline{\sigma}}(\mathbf{x}, t) \\ \partial_t \underline{\underline{\sigma}}(\mathbf{x}, t) &= \underline{\underline{C}}(\mathbf{x}) \underline{\underline{\epsilon}}(\mathbf{v}(\mathbf{x}, t)) \\ \mathbf{v}(0, \mathbf{x}) &= \mathbf{v}_0(\mathbf{x}) \\ \underline{\underline{\sigma}}(0, \mathbf{x}) &= \underline{\underline{\sigma}}_0(\mathbf{x}) \end{cases} \quad (3.120)$$

en choisissant pour conditions initiales $\underline{\underline{\sigma}}_0 = 0$ et :

$$\begin{cases} \mathbf{v}_0(\mathbf{x}) &= \begin{pmatrix} (X \cos^2 \theta + Z \sin^2 \theta)(x - x_0) + (X - Z) \cos \theta \sin \theta (z - z_0) \\ (X - Z) \cos \theta \sin \theta (x - x_0) + (Z \cos^2 \theta + X \sin^2 \theta)(z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \\ &\text{pour ne générer qu'une onde P,} \\ \mathbf{v}_0(\mathbf{x}) &= \begin{pmatrix} -(X - Z) \cos \theta \sin \theta (x - x_0) - (Z \cos^2 \theta + X \sin^2 \theta)(z - z_0) \\ (X \cos^2 \theta + Z \sin^2 \theta)(x - x_0) + (X - Z) \cos \theta \sin \theta (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2}, \\ &\text{pour ne générer qu'une onde S.} \end{cases} \quad (3.121)$$

De la même manière, pour ne générer que des ondes P dans un matériau TTI elliptique 3D, il suffit de résoudre le système de l'élastodynamique (3.117) non homogène avec pour source :

$$\underline{\underline{s}}(\mathbf{x}) = \delta_{\mathbf{x}_0}(\mathbf{x}) R_{(\theta, \phi)}^T M_{VTI, P} R_{(\theta, \phi)} \quad (3.122)$$

où $R_{(\theta, \phi)}$ désigne la matrice de changement de base entre le repère cartésien et le repère TTI 3D, définie par l'équation (1.24) et

$$M_{VTI, P} = \begin{pmatrix} \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 & 0 \\ 0 & \sqrt{\kappa^2 V_p^2 - V_s^2} & 0 \\ 0 & 0 & \sqrt{V_p^2 - V_s^2} \end{pmatrix}$$

ou en choisissant pour le problème de Cauchy (3.120) la donnée initiale :

$$\begin{cases} \underline{\underline{\sigma}}_0(\mathbf{x}) &= 0 \\ \mathbf{v}_0(\mathbf{x}) &= R_{(\theta, \phi)}^T M_{VTI, P} R_{(\theta, \phi)} \begin{pmatrix} (x - x_0) \\ (y - y_0) \\ (z - z_0) \end{pmatrix} e^{-\pi^2 \|\mathbf{x} - \mathbf{x}_0\|^2} \end{cases} \quad (3.123)$$

Théorème 3.4 Les problèmes mixtes continu, semi-discret et discret obtenus en couplant l'élastodynamique avec la CLA TTI 2D (3.115) ou 3D (3.116) sont stables au sens suivant :

– pour le problème continu, nous définissons la forme quadratique :

$$\mathbb{E}(t) = \frac{1}{2} \left(\|\mathbf{v}\|_{\Omega, \rho}^2 + \|\underline{\underline{\sigma}}\|_{\Omega, \underline{\underline{C}}^{-1}}^2 \right) \quad (3.124)$$

– pour le problème semi-discret, nous définissons la forme :

$$\mathbb{E}_{semi_d}(t) = \langle \partial_t M_v \mathbf{v}_h, \mathbf{v}_h \rangle + \langle \partial_t M_{\sigma, C^{-1}} \boldsymbol{\sigma}_h, \boldsymbol{\sigma}_h \rangle \quad (3.125)$$

– pour le problème mixte discret, nous définissons la forme :

$$\mathbb{E}_d^n = \frac{1}{2} \langle M_v \mathbf{v}_h^n, \mathbf{v}_h^n \rangle + \frac{1}{2} \langle M_{\sigma, C^{-1}} \boldsymbol{\sigma}_h^{n+\frac{1}{2}}, \boldsymbol{\sigma}_h^{n-\frac{1}{2}} \rangle \quad (3.126)$$

Alors, (3.124) et (3.125) sont toujours décroissantes au cours du temps et (3.126) est positive et décroissante si et seulement si la matrice

$$M_v - \frac{\Delta t^2}{4} R_{\sigma} (M_{\sigma, C})^{-1} (R_{\sigma})^T \quad (3.127)$$

est positive.

Corollaire 3.4.1 *Le problème mixte discret est stable sous la condition de type CFL*

$$1 - \frac{\Delta t^2}{4} \lambda_{max} \geq 0 \quad (3.128)$$

où λ_{max} est la plus grande valeur propre de la matrice $M_v^{-1} R_{\sigma} (M_{\sigma, C^{-1}})^{-1} (R_{\sigma})^T$.

Conclusion à la partie I

La simulation de la propagation des ondes élastiques anisotropes est décrite par des équations linéaires. Elle diffère du cas isotrope par le nombre de coefficients non-nuls et indépendants de son tenseur d'élasticité, sa traduisant par la présence d'intégrales supplémentaires dans les équations. L'anisotropie du sous-sol est communément considérée comme TTI (*Tilted Transverse Isotropy*), ce qui signifie que les vitesses de propagation des ondes varient selon les directions spatiales, avec des fronts d'ondes admettant une symétrie par rapport à un axe incliné. La symétrie seule implique seulement une modification des coefficients existants dans le tenseur d'élasticité isotrope, les rendant plus indépendants mais conservant la structure creuse du tenseur. L'inclinaison, quant à elle, entraîne la transformation complète de la structure du tenseur, qui devient alors dense, c'est-à-dire que tous les coefficients deviennent non-nuls, comme dans un cas d'anisotropie totale. Pour autant, cette complexité est essentiellement algébrique car tous les coefficients ne sont pas indépendants.

D'un point de vue numérique, l'introduction de conditions aux limites permettant de simuler la propagation d'une onde au-delà du domaine discret pose problème dans le cas TTI. La solution la plus commune repose sur l'utilisation de couches parfaitement adaptées (PML - *Perfectly Matched Layers*), mais elles sont mathématiquement instables. Nous avons choisi dans cette thèse de construire une Condition aux Limites Absorbantes (CLA) pour un milieu TTI.

Une méthodologie efficace et rigoureuse a été proposée par Engquist et Majda. Elle passe par le calcul des valeurs propres de la matrice associée à la propagation des ondes dans une direction donnée. Elle peut être directement appliquée au système de l'élastodynamique anisotrope et elle met en évidence des termes de couplage entre les ondes P et les ondes S. Ces termes relèvent de la physique et se traduisent, dans le meilleur des cas, par des valeurs propres faisant intervenir des opérateurs pseudo-différentiels, rendant délicate mais encore possible leur approximation. Les cas plus généraux comme l'anisotropie TTI sont plus complexes et les valeurs propres sont difficiles à calculer, même par un logiciel de calcul formel. Le plus souvent, elles ne sont pas assez explicites pour pouvoir être exploitées.

La solution que nous avons développée repose sur l'utilisation de formules géométriques sous quelques hypothèses permettant de simplifier un peu le problème :

- La première hypothèse consiste à supposer qu'une CLA peut s'exprimer en sommant les contributions des différentes ondes. Nous proposons ainsi de séparer les ondes P et les ondes S en fixant l'une ou l'autre des vitesses à zéro dans le système de l'élastodynamique. Cette approche est souvent retenue en géophysique où l'onde P est souvent la seule à être calculée avec des vitesses d'ondes S fixées selon des critères de stabilité numérique, indépendamment de leurs valeurs réelles.
- La seconde hypothèse consiste à dire que le milieu est décrit par une anisotropie TTI

de type elliptique, c'est-à-dire lorsque les premiers coefficients de Thomsen sont égaux (quels que soient les angles). Nous espérons que cette hypothèse n'a aucun impact sur la CLA d'ordre faible TTI comme c'est le cas dans un milieu VTI non-elliptique.

Sous ces deux hypothèses, la méthode d'Engquist et Majda n'est toujours pas facile à mettre en œuvre. Cependant, l'étude des surfaces de lenteur devient possible, l'ellipticité conduisant à des surfaces de formes facilement paramétrables. Il s'agit de sphères et d'ellipsoïdes inclinés. Nous avons choisi de nous concentrer sur le cas des ondes P, la CLA en ondes S étant alors la même qu'en isotrope. Dès lors, une succession de changements de variables permet de formuler une CLA TTI elliptique pour les ondes P à partir de la CLA du cas isotrope. Il suffit alors d'appliquer la somme des CLA pseudo-acoustiques pour les ondes P et S pour obtenir une CLA élastique. Ceci se justifie car un milieu TTI n'est qu'un milieu VTI incliné et la somme des CLA des problèmes découplés dans un milieu VTI est la même que la CLA obtenue sur le problème non-découplé VTI, quand il s'agit d'une CLA d'ordre peu élevé.

La stabilité mathématique des problèmes mixtes continu et semi-discret a été prouvée. La stabilité numérique du problème discret dépend d'une CFL et a été vérifiée en temps long. Les tests de validation, sur des cas TTI généraux (i.e. non-elliptique) ont mis en évidence la supériorité de cette CLA par rapport à des solutions plus simplistes (comme une CLA isotrope) et surtout, ont permis la réalisation de simulation dans des configurations où les PML sont instables.

Ces travaux ont été présentés dans plusieurs conférences internationales : Acoustics2012, ICOSAHOM2012, Waves2013 et ECCOMAS2014 notamment. Un résumé de la partie TTI 2D a été accepté pour publication : [BBCD14], la partie 3D va être soumise prochainement. D'un point de vue industriel, la CLA a été intégrée dans le code DIVA de Total où elle est utilisée pour les simulations TTI. Enfin, cette CLA a été utilisée avec succès dans les autres codes de l'équipe, reposant sur des formulations au deuxième ordre de l'équation des ondes, en temporel et en fréquentiel.

Pour aller plus loin, il est possible d'améliorer la CLA en diminuant les réflexions restantes. Pour cela, il faut considérer des développements limités à des degrés supérieurs. Les termes supplémentaires requièrent alors l'utilisation de fonctions auxiliaires dans le schéma numérique, ce qui n'est pas forcément une bonne chose pour les performances du code parallélisé. Une autre façon d'améliorer la CLA est d'ajouter une couche faiblement absorbante autour du domaine. C'est ce que l'on appelle un "taper". Cette solution est efficace mais conduit au même désavantage que les PML, à savoir l'obligation de faire des calculs subsidiaires dans la zone d'intérêt, ce qui augmente de façon non-négligeable le temps de simulation, surtout en 3D.

Nous envisageons plusieurs suites possibles à ces travaux. Dans un premier temps, nous souhaitons comparer la CLA avec les couches dites SMART d'Halpern et al. [HPBR11] et Métivier et al. [MBL+14]. Elles utilisent une couche autour du maillage permettant de traiter le problème TTI acoustique. Cela nécessite donc d'abord une extension des SMART vers l'élastodynamique TTI. Ensuite, nous aimerions coupler la CLA et la couche SMART en posant la CLA au bord du domaine afin de limiter la taille de la couche. Le tout pourrait aussi être comparé à l'utilisation d'un "taper". Une différence importante entre la CLA et les couches SMART est que la CLA s'applique facilement à des formulations du premier et du

second ordre alors que les couches SMART ne s'appliquent pour l'instant qu'aux systèmes du second ordre.

Il nous semble aussi intéressant de comparer le comportement des CLA entre les systèmes du premier et du second ordre, numériquement mais également d'un point de vue HPC. Enfin, nous travaillons déjà sur l'étude détaillée des CLA TTI 3D, avec notamment le calculs des coefficients de réflexion pour les ondes P et le découplage des ondes S qui permettrait en suivant notre méthodologie de construire une CLA lorsque $\gamma \neq 0$ et d'analyser les coefficients de réflexion pour les ondes S.

Deuxième partie

Programmation à base de tâches
sur support d'exécution

Introduction à la partie II

Les supercalculateurs, interconnexions d'ordinateurs, sont aujourd'hui un outil incontournable de la simulation numérique. La plupart des laboratoires d'ingénierie, public ou privé, possèdent le leur ou en louent un accès. Cette démocratisation a débuté avec l'ère des machines massivement parallèles à la fin des années 1980. À cette époque, les évolutions sur les processeurs se traduisaient directement en gain de performance sur l'exécution des programmes informatiques. Citons par exemple l'augmentation exponentielle du nombre de transistors ou de la fréquence d'horloge, mais aussi la complexité grandissante du parallélisme des instructions de l'unité de contrôle (en anglais ILP, *Instruction-Level Parallelism*), ou encore l'augmentation de la taille des niveaux de mémoire cache.

Cette évolution a atteint ses limites au cours des années 2000 (voir Fig. 3.30) en se heurtant principalement au “mur thermique”. En effet, la miniaturisation des composants, associée à des fréquences d'horloge très élevées, ne permettait plus une bonne dissipation de la chaleur. À cela s'est ajouté un problème de consommation électrique, trop directement proportionnelle à la puissance de calcul, précipitant la fin de l'architecture processeur mono-cœur.

Néanmoins, le nombre de transistors a continué à évoluer selon la loi de Moore¹, suivant un nouveau type d'architecture intrinsèquement parallèle : multi-cœurs puis *hyperthreading* (voir Fig. 3.31). Dès lors, exploiter efficacement les ressources de ces machines va requérir un travail spécialisé, marquant une rupture avec l'ère précédente, comme l'avait observé Herb Sutter en 2005 dans son article au titre évocateur, “*The free lunch is over*” [Sut05].

Cette architecture parallèle est logiquement devenue prédominante dans les supercalculateurs. Cependant, comme la fréquence d'horloge n'augmentait plus, et que les optimisations de l'ILP et de la mémoire cache n'étaient plus aussi marquées, il fallait dorénavant beaucoup plus de processeurs à chaque renouvellement de machine, pour obtenir les mêmes marges de progression qu'au cours des années précédentes.

La Fig. 3.32 représente l'évolution des supercalculateurs selon le classement du top500². La puissance de calcul effective est exprimée en Flops (de l'anglais *FL*loating-*point* *O*perations *P*er *S*econd), mesurée sur le benchmark de référence LINPACK³. Malgré le changement brutal d'architecture, les performances ont continué à croître de manière continue.

1. La loi de Moore, dérivée d'un article de 1965 [Moo65], prévoit que le nombre de transistors des processeurs double tous les deux ans. Les constructeurs s'en sont rapidement servi comme feuille de route technologique.

2. Le top500 [DMS97] est un recensement tous les six mois des 500 supercalculateurs les plus puissants. C'est la référence historique et toujours actuelle des tendances et évolutions pour la communauté HPC.

3. Le benchmark LINPACK [Don88], introduit par Jack Dongarra en 1979, mesure le temps de résolution d'un système matriciel dense par une méthode directe. Ce test de performance historique devrait prochainement évoluer en considérant une résolution matricielle creuse par une méthode itérative [DH13].

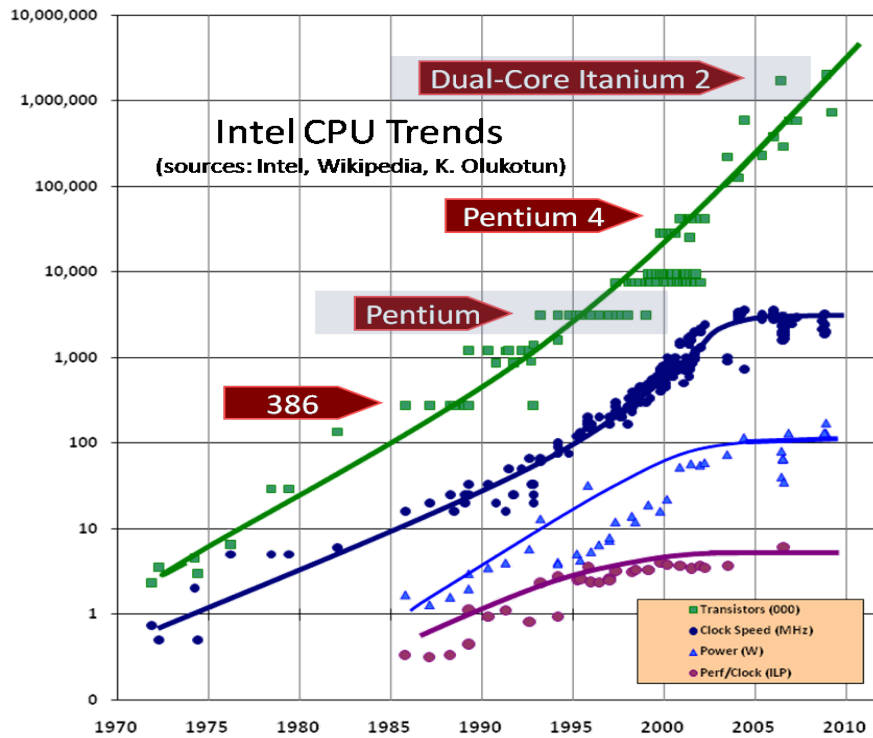


FIGURE 3.30 – Évolution des processeurs Intel

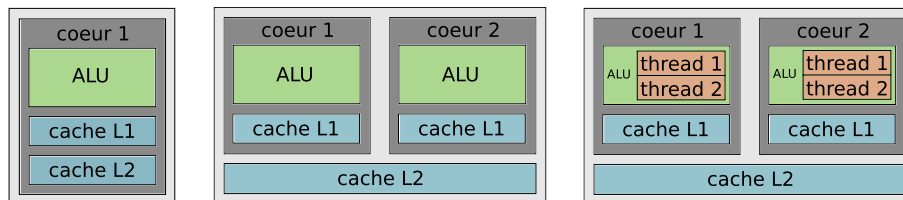


FIGURE 3.31 – Architectures des processeurs, mono-cœur (gauche), multi-cœurs (centre) et *hyperthreading* (droite)

Depuis quelques années, l'évolution des supercalculateurs est confrontée à une nouvelle difficulté. L'augmentation exponentielle du nombre de processeurs fait exploser la consommation électrique, dont une partie importante est dédiée au refroidissement des machines. Sur ce point, un nouveau classement de performance par watt, le Green500, a été établi afin de mettre en avant l'efficacité énergétique. Au-delà d'un problème écologique et économique, l'utilisation simultanée d'un très grand nombre de cœurs pose des problèmes d'accès efficaces aux données, ce qui détériore le passage à l'échelle des performances effectives.

Pour pallier en partie ce problème de consommation tout en augmentant la puissance de calcul, la communauté HPC s'est tournée à la fin des années 2000 vers les accélérateurs. Ces cartes annexes, qui s'ajoutent aux nœuds existants des supercalculateurs, ont permis

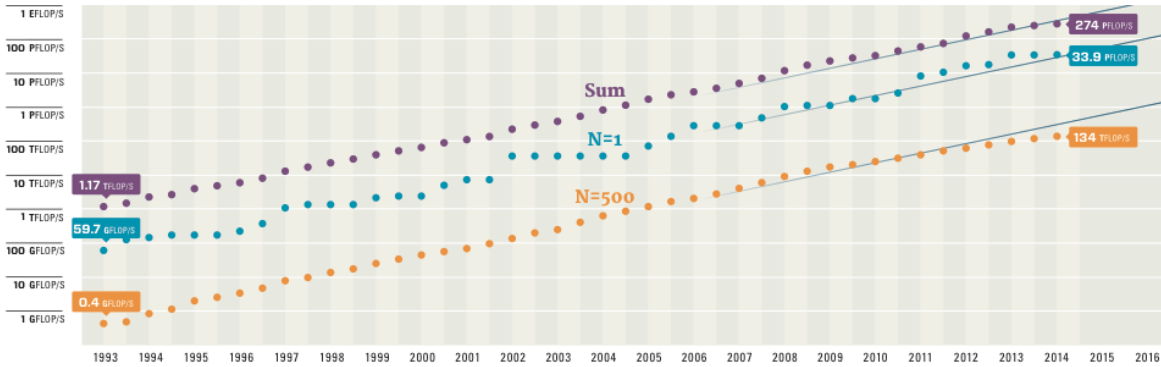


FIGURE 3.32 – Evolution des supercalculateurs du top500

d'augmenter les performances théoriques par watt (et par dollar). Il convient de distinguer leurs architectures :

- Les GPGPU (*General Purpose on Graphic Processing Units*), font référence à des cartes graphiques détournées pour la simulation numérique. Ils possèdent une architecture particulière, les unités de calculs y sont très nombreuses, très spécialisées et se partagent quelques unités de contrôle très simplifiées (voir Fig. 3.33). Leur utilisation requiert de réécrire les codes avec un langage de programmation spécifique. Le parallélisme est à exprimer sur une grille virtuelle et les optimisations sont souvent complexes. Dans le reste du manuscrit, nous utiliserons le terme GPU, faisant implicitement référence au GPGPU.
- Les coprocesseurs sont quant à eux une modification des processeurs classiques pour le calcul intensif. Il sont apparus un peu plus tard, au début des années 2010. Leur architecture est une évolution du modèle multi-cœurs vers un modèle “many-cœurs”, comme celle de l’Intel Xeon Phi appelée MIC (*Many-Integrated Cores*, voir Fig. 3.33). Les cœurs sont plus nombreux que sur les processeurs multi-cœurs. Le contrôle de flot est plus simple et compensé par un support matériel de l’*hyperthreading*. Leur utilisation est simplifiée par rapport aux GPU car elle fait appel aux mêmes outils que sur les CPU. Toutefois, leur utilisation optimisée reste un défi complexe.

Dans la pratique, c’est aux développeurs que revient le soin de répartir le travail sur les ressources disponibles (processeurs standards et accélérateurs) en fonction de leur capacité de traitement : nombre d’unités de calcul, puissance de ces unités (en fonction du degré d’ILP et de la présence d’*hyperthreading*), caractéristiques des différents niveaux de mémoire de chacune des architectures. Programmer de telles machines hétérogènes a amené Herb Sutter à écrire en 2012 un nouvel article [Sut12] dont le titre, “*Welcome to the jungle*”, évoque toute la complexité de la tâche.

Ce caractère hétérogène des ressources informatiques continue aujourd’hui de s’amplifier. En effet, la miniaturisation tend à atteindre l’échelle atomique et pourrait d’ici quelques dizaines d’années stopper sa progression. Suivant la loi de Moore, l’architecture des processeurs standards commence déjà à s’orienter vers celle des *many-cœurs*. Citons par exemple la version processeur des Intel Xeon Phi (“Knights Landing”), qui sera disponible l’année prochaine, ou, plus orienté basse consommation, le MPPA de Kalray qui propose déjà une alternative similaire sur un autre jeu d’instructions.

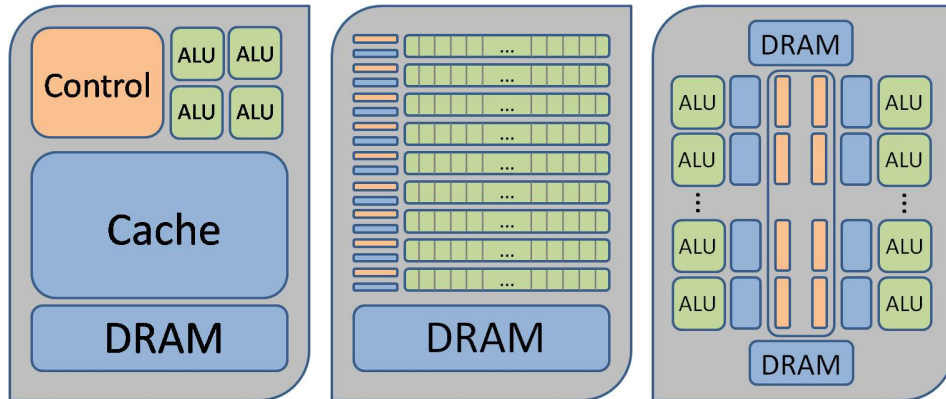


FIGURE 3.33 – Architectures CPU (gauche), GPU (centre) et MIC (droite)

En ajoutant à cela la course à l'*exascale* [DBea11] (c'est-à-dire atteindre l'exaFlops : 10^{18} Flops), se traduisant par la gestion de milliards d'unités de calculs, la problématique d'avoir des codes performants, portables et flexibles devient incontournable et centrale. Or, les programmes informatiques, lorsqu'ils sont optimisés pour des machines hétérogènes, se retrouvent dépendants du matériel, au détriment potentiel d'évolutivité.

Ce constat met en évidence une limite potentielle des modèles de programmation dominants actuels : il faut maintenant les composer pour exploiter plusieurs niveaux de parallélisme. Par exemple, pour programmer une machine composée de nœuds accélérés par des GPU, le modèle le plus répandu permettant l'utilisation de ce type d'architecture hétérogène est le triptyque MPI+*thread*+CUDA. MPI [MP198] (*Message Passing Interface*) se charge de la communication entre les différents nœuds, le niveau *thread* (par exemple POSIX ou OpenMP) est spécialisé pour gérer plus efficacement les différents cœurs d'un même nœud et CUDA (ou OpenCL) est utilisé pour exploiter les GPU.

Sur la base de ces modèles, demander aux développeurs de la performance sur une machine hétérogène nécessite alors beaucoup de temps et d'expertise, notamment pour la gestion des transferts de données et la répartition équilibrée des calculs. De plus, à chaque évolution architecturale, une grande partie du travail d'optimisation est à recommencer.

La programmation à base de tâches, à l'aide de supports d'exécution (en anglais *task-based runtime system*), permet de limiter la dépendance du code à l'architecture sous-jacente. Le code est décrit par des tâches, sans lien direct *a priori* avec le matériel. Le support d'exécution se charge ensuite d'exécuter ces tâches sur les ressources disponibles, tout en assurant la cohérence des données, c'est-à-dire l'assurance d'utiliser une copie d'une donnée pour laquelle toutes les modifications antérieures sur d'autres copies de cette même donnée ont été appliquées. Pour être efficaces, ils mettent en place des politiques d'ordonnancement avancées (statiques ou dynamiques). Certains d'entre eux peuvent aussi se charger des transferts de données distantes de façon transparente, comme nous le verrons à la section 5.2.

Un programme peut ainsi être modélisé par un graphe dont les nœuds représentent les tâches élémentaires et les arêtes décrivent les dépendances entre ces tâches. Écrire un code performant revient alors à construire un graphe de tâches exprimant beaucoup de parallélisme, c'est-à-dire dont les dépendances entre les tâches sont limitées. Optimiser ce code pour une

architecture donnée consiste dès lors principalement à réécrire les tâches élémentaires selon les spécificités du matériel (si besoin) et à choisir une politique d’ordonnancement permettant une adéquation entre l’application et l’architecture. Les supports d’exécution fournissent souvent des algorithmes d’ordonnancement prédéfinis qu’il suffit généralement de paramétrer convenablement. Cependant, il se peut qu’en passant d’une architecture à une autre, le parallélisme fourni par le graphe de tâches ne convienne plus. Dans ce cas, il est nécessaire d’en affiner son expression.

L’objectif de cette deuxième partie consiste alors à mettre au point une approche par tâches du code de propagation des ondes élastiques de Total, puis à utiliser un support d’exécution, en l’occurrence PaRSEC (voir section 5.3). Le plan se décompose en quatre chapitres.

Le chapitre 4 décrit l’algorithme de propagation des ondes du code DIVA de Total. La section 4.1 donne le pseudo-code parallèle par passage de messages de l’implémentation initiale avec MPI. Une étude de performance est présentée 4.2 pour illustrer la problématique de ce modèle basé sur une décomposition de domaines. Le chapitre 5 fait une présentation générale du paradigme de parallélisme par tâches et des supports d’exécution servant à ordonnancer ces tâches. En particulier, la section 5.1 détaille comment porter un exemple simple en tâches sur deux modèles différents. La section 5.2 dresse un état de l’art des supports d’exécution pour la programmation en tâches et la section 5.3 résume les spécificités de celui que nous avons utilisé : PaRSEC.

Le chapitre 6 est consacré à la description du portage du code DIVA en tâches. La section 6.1 reprend ainsi les différentes étapes décrites au chapitre précédent pour le code DIVA. Puis, la section 6.2 explique comment exploiter efficacement les architectures à mémoire partagée avec une illustration des points clés sur un processeur. Le chapitre 7 présente l’étude expérimentale comparative entre l’implémentation initiale en MPI et le code porté en tâches. Le cas test est décrit à la section 7.1 et les résultats sont développés à la section 7.2, notamment pour des machines ccNUMA et Intel Xeon Phi.

Dans le cadre de l’action stratégique DIP, ce travail a été réalisé au sein d’une collaboration avec George BOSILCA de l’ICL (*Innovative Computing Laboratory, University of Tennessee, USA*), développeur en chef de PaRSEC, sous la coordination d’Emmanuel AGULLO chargé de recherches Inria dans l’équipe *Hiepac*s. George a passé plusieurs mois en France, ce qui a facilité les échanges directs et intensifs sur le projet.

Commençons par rappeler quelques définitions fondamentales du calcul parallèle auxquelles nous allons nous référer dans ce manuscrit.

Efficacité

L’efficacité d’un code parallèle traduit l’utilisation adéquate des ressources disponibles. C’est une valeur comprise entre zéro et un. Plus cette valeur est proche de 1, plus le code utilise les capacités maximales de la machine. Soit p le nombre de ressources parallèles et T le temps. T_s représente le temps de référence pour une exécution sur une seule unité de calcul, appelé temps séquentiel, et T_p le temps d’exécution sur p unités de calcul, appelé temps

parallèle. L'efficacité est alors définie par la formule :

$$E(p) = \frac{T_s}{p.T_p} \quad (3.129)$$

Speedup

Pour mesurer le gain éventuel de la modification d'un code, le rapport du temps d'exécution obtenu avec le code initial, noté $T_{initial}$, sur le temps d'exécution obtenu avec le code final, noté T_{final} , est appelé *speedup*. C'est une valeur positive. Lorsque cette valeur est plus grande que un, le code final est plus rapide que le code initial. En programmation parallèle, il existe aussi un indicateur *speedup* obtenu cette fois par le rapport du temps séquentiel sur le temps parallèle (T_s/T_p). C'est la première définition qui est utilisée dans cette thèse. Ainsi, la formule du *speedup* S utilisée est :

$$S = \frac{T_{initial}}{T_{final}} \quad (3.130)$$

Passage à l'échelle

Dérivé de l'anglais *scalability*, le passage à l'échelle correspond à l'adéquation entre le temps d'exécution d'une expérience et le nombre de ressources parallèles utilisées. Il est d'usage de différencier deux définitions :

- scalabilité forte : les expériences concernent toujours le même cas test (à taille fixée). Dans ce cas, le temps d'exécution idéal diminue autant que le nombre de ressources augmente, par exemple pour deux fois plus de ressources, le temps d'exécution idéal est divisé par deux.
- scalabilité faible : la taille des cas tests varie avec le nombre de ressources, afin de garder une charge de travail constante pour chaque ressource quel que soit le cas test. Cette fois, le temps d'exécution idéal reste le même, par exemple pour deux fois plus de ressources et un cas test deux fois plus gros.

Granularité

Diminuer la taille des tâches élémentaires d'un programme permet d'exhiber potentiellement davantage de parallélisme, mais en contrepartie, leur surcoût de traitement peut augmenter. La granularité représente le rapport entre la taille des tâches et leur surcoût de traitement. Ainsi, optimiser la granularité consiste à déterminer un compromis entre un découpage en beaucoup de petite tâches (grain plus fin) et la gestion de leur surcoût de traitement en limitant le nombre et donc la taille des tâches (grain plus grossier).

La Fig. 3.34 illustre ce compromis. Chacun des quatre schémas représente la même quantité de calculs : la surface pointillée blanche, qui est découpée en 1, 4, 16 et 64 tâches de gauche à droite. Le surcoût de traitement est représenté en noir. Schématiquement, il dépend de la taille des tâches tout en conservant une largeur fixe. Sur le schéma de gauche, il n'y a pas de parallélisme, le temps d'exécution correspond à la surface du schéma, c'est le temps séquentiel. Sur les autres schémas, le parallélisme est de plus en plus important, jusqu'à obtenir tout à droite, des tâches si petites que leur surcoût de traitement est largement supérieur à leur quantité de calculs.

Pour un grand nombre de ressources parallèles, il peut être intéressant de tendre vers un grain très fin et d'attribuer chaque petite tâche à une ressource. Cependant, l'efficacité peut ne pas être bonne si l'essentiel du temps d'exécution est dédié au surcoût de traitement (cela se vérifie en comparant la surface totale des schémas). Le passage à l'échelle peut ne pas être assuré non plus, puisque le temps d'exécution peut augmenter plus vite que le nombre de ressources (cela se vérifie en comparant la surface d'une tâche sur chaque schéma).

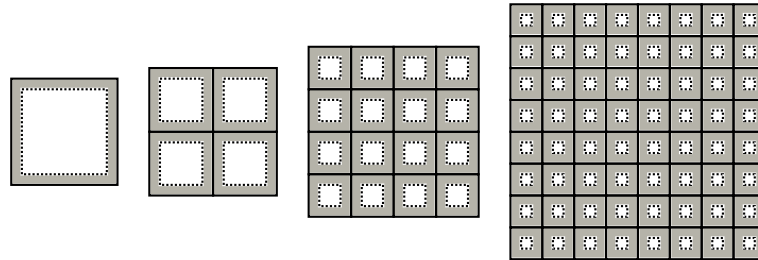


FIGURE 3.34 – Illustration des effets potentiels de la granularité

Équilibrage de charge

L'exécution d'un code parallèle se décompose en calculs et en communications ou synchronisations. Lorsque les calculs ne prennent pas le même temps sur chaque ressource, les ressources les moins chargées peuvent être amenées à attendre les plus chargées pour pouvoir poursuivre l'exécution du code. L'équilibrage de charge, en l'anglais *load balancing*, mesure cette répartition des calculs, ce qui a pour objectif de limiter ces temps d'attentes. La Fig. 3.35 illustre l'équilibrage de charge possible sur deux ressources parallèles, pour autant que la granularité soit suffisamment fine. Les blocs contiennent les tâches et leur surcoût de traitement.

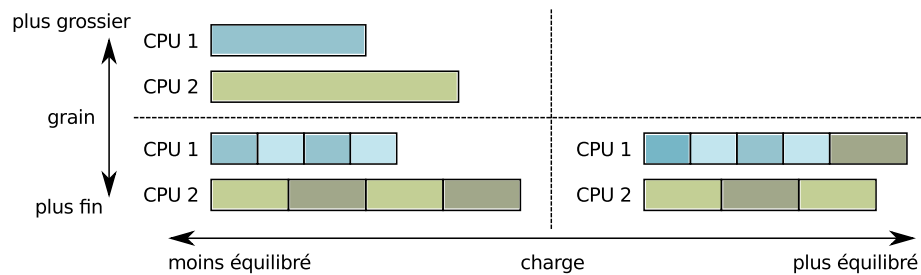


FIGURE 3.35 – Illustration de l'équilibrage de charge sur deux CPUs, grain fin et grain grossier

Chapitre 4

Étude du code DIVA initial pour la propagation des ondes

Ce chapitre décrit l'algorithme de propagation des ondes utilisé dans le code DIVA. Les équations physiques ont été discrétisées en un système matriciel et nous commençons par présenter l'algorithme séquentiel correspondant. Nous détaillons ensuite la parallélisation qui repose sur une décomposition du domaine d'étude et l'utilisation du paradigme de parallélisme par passage de messages. Pour finir, nous étudions les propriétés du code parallèle en illustrant son comportement sur une machine ciblée. Ceci va nous permettre d'identifier le problème majeur de l'implémentation initiale et notre objectif sera de le corriger.

4.1 Implémentation par passage de message

4.1.1 Pseudo-code séquentiel à partir de la discrétisation de la partie I

Dans la première partie de cette thèse, nous avons expliqué comment le système de l'élastodynamique est discrétisé (voir section 1.3). Nous utilisons une méthode d'éléments finis DG en espace et un schéma Leap-Frog en temps. Ce choix favorise le parallélisme en découpant le travail maille par maille. Les matrices de masse M_\star et B_\star ont alors une structure diagonale par blocs et les matrices de rigidité R_\star ont un *stencil* fixe (seulement les données des mailles directement voisines par les faces sont nécessaires). En contrepartie, le nombre de degrés de liberté total est plus élevé que dans les méthodes d'éléments finis continus. Nous pouvons toutefois remarquer que de nouvelles méthodes DG dites hybrides (voir [CGL09] pour l'article de référence et [BGCDL14] pour une application à l'élastodynamique) commencent à émerger et nous pouvons observer qu'elles conservent les mêmes avantages que les méthodes DG tout en utilisant un nombre de degrés de liberté raisonnable et comparable à celui requis par les méthodes continues.

Le maillage est non structuré, composé de N_h mailles K . Les inconnues vecteur vitesse \mathbf{v} et tenseur des contraintes $\underline{\sigma}$ sont approchées par des vecteurs, notés \mathbf{v}_h et $\boldsymbol{\sigma}_h$ respectivement. L'intervalle de temps $[0, T]$ est divisé en N_t pas de temps Δt . En notant \mathbf{v}_h^n l'approximation \mathbf{v}_h au temps discret $t = n\Delta t$ et $\boldsymbol{\sigma}_h^{n+1/2}$ celle de $\boldsymbol{\sigma}_h$ au temps discret intermédiaire $t = (n + \frac{1}{2})\Delta t$,

le système discret s'écrit :

$$\begin{cases} M_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} - \mathbf{v}_h^n}{\Delta t} + R_{\sigma} \sigma_h^{n+1/2} + B_{\mathbf{v}} \frac{\mathbf{v}_h^{n+1} + \mathbf{v}_h^n}{2} & = 0 \end{cases} \quad (4.1a)$$

$$\begin{cases} M_{\sigma} \frac{\sigma_h^{n+3/2} - \sigma_h^{n+1/2}}{\Delta t} + R_{\mathbf{v}} \mathbf{v}_h^{n+1} & = 0 \end{cases} \quad (4.1b)$$

Les matrices de masse M_{\star} et B_{\star} sont diagonales par blocs, chacune de taille proportionnelle au nombre de degrés de liberté de chaque maille, variant entre 4 et 20 (voir Tab. 1.2). Il est donc possible de les inverser pour un coût très faible et le système se réécrit de façon quasi-explicite :

$$\begin{cases} \mathbf{v}_h^{n+1} & = \left(\frac{1}{\Delta t} M_{\mathbf{v}} + \frac{1}{2} B_{\mathbf{v}} \right)^{-1} \left(\frac{1}{\Delta t} M_{\mathbf{v}} - \frac{1}{2} B_{\mathbf{v}} \right) \mathbf{v}_h^n - \left(\frac{1}{\Delta t} M_{\mathbf{v}} + \frac{1}{2} B_{\mathbf{v}} \right)^{-1} R_{\sigma} \sigma_h^{n+1/2} \\ \sigma_h^{n+3/2} & = \sigma_h^{n+1/2} - \Delta t M_{\sigma}^{-1} R_{\mathbf{v}} \mathbf{v}_h^{n+1} \end{cases} \quad (4.2)$$

Pour simplifier la présentation de la méthode, nous allons considérer des conditions aux limites triviales $B_{\mathbf{v}} = 0$, ce qui correspond physiquement à une condition de surface libre. Ce ne sont pas les conditions aux limites correspondant à l'utilisation habituelle du code DIVA car nous avons décidé de nous concentrer dans un premier temps sur le cœur des calculs. Le problème des conditions aux limites est un sujet important qu'il est préférable de traiter dans un second temps. En effet, l'utilisation de conditions aux limites ajoute de la complexité dans l'algorithme et, selon le choix de ces conditions, le comportement du code peut être très différent. Cependant, ce point fait partie de nos perspectives de travail à court terme.

Le système que nous considérons est alors le suivant :

$$\begin{cases} \mathbf{v}_h^{n+1} & = \mathbf{v}_h^n - \Delta t M_{\mathbf{v}}^{-1} R_{\sigma} \sigma_h^{n+1/2} \end{cases} \quad (4.3a)$$

$$\begin{cases} \sigma_h^{n+3/2} & = \sigma_h^{n+1/2} - \Delta t M_{\sigma}^{-1} R_{\mathbf{v}} \mathbf{v}_h^{n+1} \end{cases} \quad (4.3b)$$

Le pseudo-code séquentiel associé à la résolution de ce système correspond à l'algorithme 4.1. Les équations (4.3a) et (4.3b) sont représentées par des fonctions de mise à jour des variables, lignes 4 et 7 respectivement. La discrétisation se traduit par la présence d'une boucle en temps, ligne 2, et d'une boucle en espace pour chaque équation, lignes 3 et 6. Les calculs sont effectués maille par maille par les fonctions de mise à jour qui nécessitent les deux variables \mathbf{v}_h et σ_h au pas de temps courant pour l'une et au demi-pas de temps précédent pour l'autre. De plus, les matrices de rigidité R_{\star} font intervenir les valeurs locales mais aussi directement voisines de la variable prise au demi-pas de temps précédent. Pour une maille K , cet ensemble de valeurs situées sur les degrés de liberté des faces des voisins est noté par la suite \bar{K} (par analogie avec l'adhérence, voir Fig. 4.1). Ainsi, les deux mises à jour des variables ne peuvent être faites dans une seule boucle en espace.

4.1.2 Pseudo-code parallèle par passage de messages

Le parallélisme repose sur la décomposition du domaine spatial en sous-domaines, sans recouvrement. La Fig. 4.2 illustre un exemple de découpage sur un maillage 2D. Les pointillés

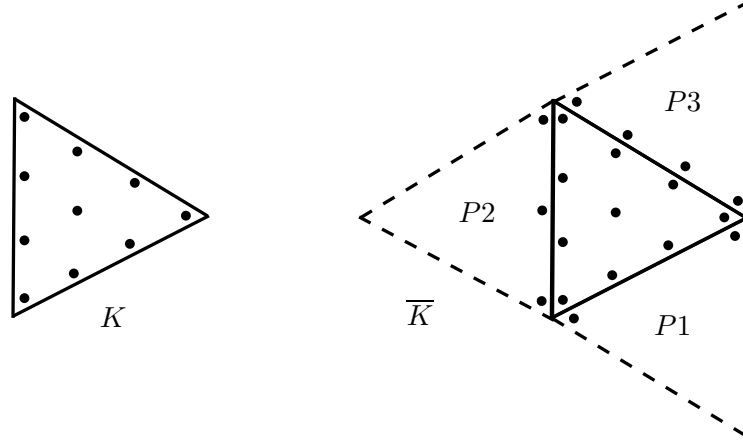


FIGURE 4.1 – Ensemble \bar{K} des degrés de liberté qui interviennent dans la matrice de rigidité locale à la maille K sur un exemple 2D où l'ordre de discrétisation des mailles voisines n'est pas homogène.

Algorithme 4.1 : Pseudo-code séquentiel de DIVA (discrétisation DG + Leap-Frog)

Data : N_h, Δ_t, N_t
Result : $\mathbf{v}_h, \boldsymbol{\sigma}_h$
 1 $[\mathbf{v}_h^1, \boldsymbol{\sigma}_h^{3/2}] \leftarrow \text{Initialization}(N_h, \Delta_t);$
 2 **for** $n = 1..N_t$ **do**
 3 **for** $K = 1..N_h$ **do**
 4 $\mathbf{v}_{h_K}^{n+1} \leftarrow \text{UpdateVelocity}(\mathbf{v}_{h_K}^n, \boldsymbol{\sigma}_{h_{\bar{K}}}^{n+1/2}, \Delta_t);$
 5 **end**
 6 **for** $K = 1..N_h$ **do**
 7 $\boldsymbol{\sigma}_{h_K}^{n+3/2} \leftarrow \text{UpdateStress}(\boldsymbol{\sigma}_{h_K}^{n+1/2}, \mathbf{v}_{h_{\bar{K}}}^{n+1}, \Delta_t);$
 8 **end**
 9 **end**

représentent les données nécessaires aux calculs sur un sous-domaine voisin, correspondant aux degrés de liberté des faces au contact des interfaces. Cette décomposition constitue la première étape de la parallélisation, ligne 1 de l'algorithme 4.2 du pseudo-code parallèle de DIVA (les différences par rapport au pseudo-code séquentiel, algorithme 4.1, sont notées en bleu). L'implémentation initiale du code crée autant de sous-domaines que de ressources parallèles disponibles N_p . Une fois cette décomposition effectuée (et les données réparties sur les processus correspondants), chaque processus initialise ses variables locales, \mathbf{v}_h et $\boldsymbol{\sigma}_h$, ligne 2, sur les $N_{h_{loc}}$ mailles locales constituant son sous-domaine. La boucle en temps, ligne 3 reste inchangée tandis que les boucles en espace, lignes 5 et 9, ne s'appliquent maintenant qu'aux mailles locales au sous-domaine. Le changement le plus important concerne les mailles situées aux bords d'un sous-domaine. Les degrés de liberté des faces des voisins ne sont plus directement accessibles puisque les mailles voisines sont sur un autre sous-domaine, donc inconnues localement. C'est ici qu'interviennent les phases de communications, lignes 4 et 8. Les données situées localement aux interfaces sont envoyées et les données situées sur les faces voisines aux

interfaces sont reçues, permettant les calculs sur \bar{K} (voir Fig. 4.1). Pour finir, les fonctions de mise à jour, lignes 6 et 10, sont identiques au code séquentiel, excepté qu’elles correspondent à des calculs locaux avec des données d’entrée et de sortie appartenant au sous-domaine local.

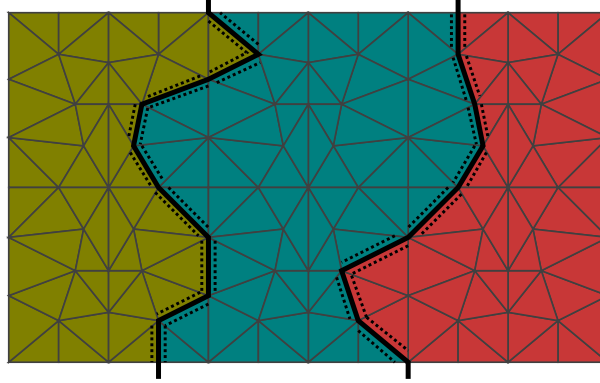


FIGURE 4.2 – Exemple de découpage d’un maillage 2D en trois sous-domaines (vert, bleu et rouge), les pointillés correspondent aux données à échanger

Algorithme 4.2 : Pseudo-code parallèle de DIVA par passage de messages

Data : N_p, N_h, Δ_t, N_t
Result : $\mathbf{v}_h, \boldsymbol{\sigma}_h$

```

1  $N_{h_{loc}} \leftarrow \text{DomainDecomposition}(N_p);$ 
2  $[\mathbf{v}_h^1, \boldsymbol{\sigma}_h^{3/2}] \leftarrow \text{Initialization}(N_{h_{loc}}, \Delta_t);$ 
3 for  $n = 1..N_t$  do
4    $\boldsymbol{\sigma}_{h_{\bar{K}}}^{n+1/2} \leftarrow \text{Communication}(\boldsymbol{\sigma}_{h_K}^{n+1/2});$ 
5   for  $K = 1..N_{h_{loc}}$  do // - numérotation interne au sous-domaine
6      $\mathbf{v}_{h_K}^{n+1} \leftarrow \text{UpdateVelocity}(\mathbf{v}_{h_K}^n, \boldsymbol{\sigma}_{h_{\bar{K}}}^{n+1/2}, \Delta_t);$ 
7   end
8    $\mathbf{v}_{h_{\bar{K}}}^{n+1} \leftarrow \text{Communication}(\mathbf{v}_{h_K}^{n+1});$ 
9   for  $K = 1..N_{h_{loc}}$  do // - numérotation interne au sous-domaine
10     $\boldsymbol{\sigma}_{h_K}^{n+3/2} \leftarrow \text{UpdateStress}(\boldsymbol{\sigma}_{h_K}^{n+1/2}, \mathbf{v}_{h_{\bar{K}}}^{n+1}, \Delta_t);$ 
11  end
12 end
```

Cet algorithme par passage de messages est composé de phases de calculs et de communications entrelacées, sans barrières de synchronisation globale. Nous appellerons ce schéma “*compute and exchange*”. Dans notre cas, les communications ne concernent qu’un volume très faible de données, à savoir les degrés de liberté des faces aux interfaces. L’essentiel du temps d’exécution correspond donc à du temps de calcul effectif. Cet algorithme est implémenté en MPI dans le code DIVA initial.

4.2 Étude de performance

S'il n'y a pas de synchronisation globale, chaque processus doit malgré tout échanger avec ses voisins avant de pouvoir entamer une nouvelle phase de calcul. L'équilibrage de charge est donc primordial pour éviter qu'un processus peu chargé attende un processus plus chargé. La décomposition de domaine étant réalisée en amont de la phase de calculs, elle se base sur une estimation *a priori* des temps de calculs. Pour assurer un équilibrage de charge cohérent, l'étape de découpage du domaine prend en compte le nombre de degrés de liberté de chaque maille. Il faut également prendre en compte le nombre de degrés de liberté des faces des mailles voisines, ce qui permet au moins de calculer le nombre exact des opérations intervenant dans les fonctions de mise à jour. Cependant, en fonction des capacités des ressources matérielles (degré d'ILP, taille des caches, ...), le nombre exact d'opérations ne donne pas le temps de calculs exact. Nous retrouvons ici le problème de l'hétérogénéité. Dans la pratique, nous nous servons de formules approximatives pour prendre en compte l'architecture sous-jacente. Obtenir un équilibrage de charge parfait est donc délicat. Un travail d'optimisation est toujours possible *a posteriori* pour améliorer l'efficacité du code mais de façon dépendante à l'architecture, donc non portable. C'est ainsi la limite de ce modèle de programmation pour le code DIVA.

La Fig. 4.3 présente une trace d'exécution de 10 itérations du code DIVA en MPI sur 32 cœurs. Les traces représentent les temps d'activité et d'inactivité de chaque ressource au cours de l'exécution, disposées en ligne. Les noyaux de calcul de la vitesse et du tenseur des contraintes sont représentés en orange et rouge respectivement. Le gris clair met en évidence les temps d'attente dus aux communications par MPI. Cet exemple montre que, même si la plupart du temps est passé à effectuer des calculs, il subsiste un léger déséquilibre de charge (certains cœurs en attendant d'autres), cela se vérifie en comparant la taille des blocs de couleurs correspondants aux mises à jour des variables. Comme le code DIVA est répétitif et que les sous-domaines restent fixes, ce phénomène se reproduit à chaque itération. Au final, beaucoup de zones grises apparaissent, traduisant une sous-utilisation potentielle des ressources.

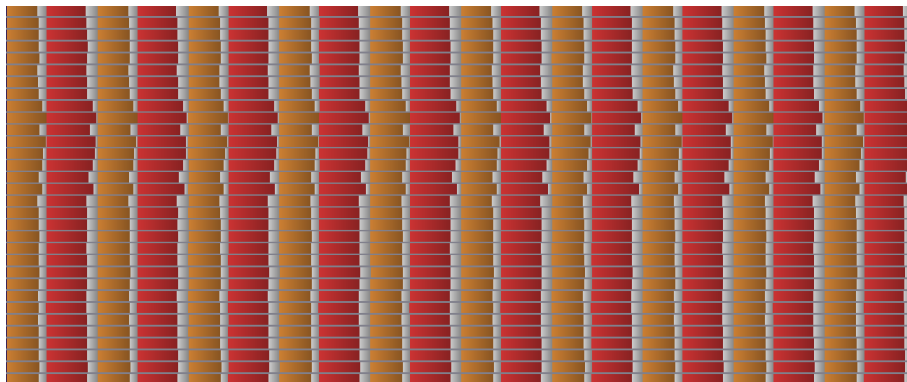


FIGURE 4.3 – Trace d'une exécution du code DIVA sur 32 cœurs (un par ligne) et 10 itérations. Les calculs sont en orange (vitesse) et rouge (contraintes) et les temps d'attente en gris clair

Nous présentons maintenant une étude de performance du code DIVA. Le cas test considéré

ici est un cube 3D hétérogène, c'est-à-dire contenant différents milieux physiques, dont les spécificités sont détaillées à la section 7.1. Nous considérons deux tailles de maillages constitués de 200.000 et 800.000 tétraèdres, générés avec Tetgen. Le nombre de degrés de liberté de chaque maille dépend de ses caractéristiques physiques locales. La machine utilisée est homogène, constituée de 8 processeurs Intel Xeon E7-8837, composé chacun de 8 CPUs cadencés à 2.67 GHz. Chaque cœur CPU possède 64 KB et 256 KB de cache L1 et L2 respectivement, et chaque processeur dispose de 24 MB de cache L3.

La Fig. 4.4 présente l'efficacité parallèle du code DIVA basé sur MPI. La figure de gauche correspond à une discrétisation homogène, c'est-à-dire avec le même nombre de degrés de liberté dans chaque maille. Dans cette configuration, l'efficacité reste bonne pour les deux tailles de maillages et ne décroît que très légèrement. La figure de droite correspond à une discrétisation hétérogène, ce qui signifie que le nombre de degrés de liberté dans chaque maille est différent. C'est la configuration classique. Dans ce cas, l'efficacité décroît au fur et à mesure que le nombre de processeurs augmente, et ce quelle que soit la taille du maillage utilisé. En effet, la connaissance du nombre exact d'opérations ne suffit généralement pas pour déduire par des formules le poids exact des calculs de chaque sous-domaine lorsque les données ne sont pas homogènes.

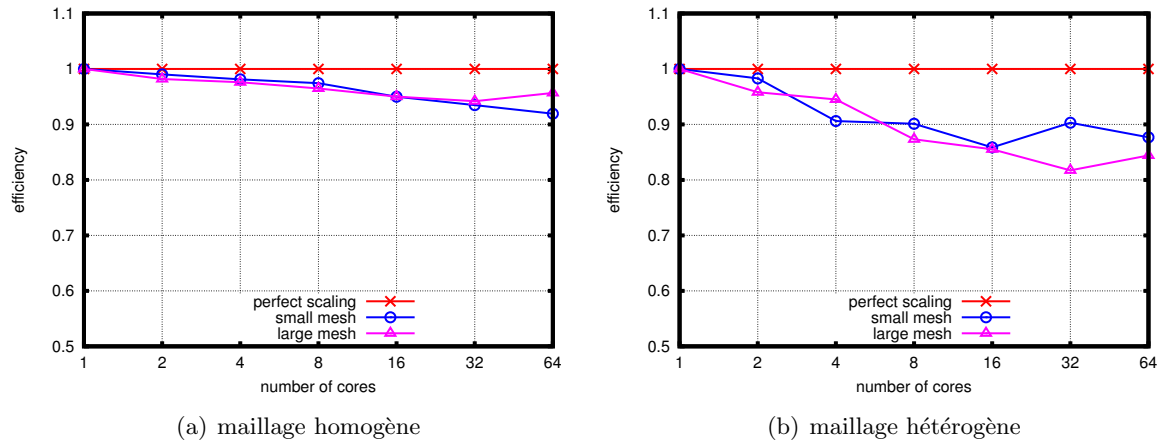


FIGURE 4.4 – Étude de l'efficacité parallèle du code DIVA

Dans la suite de ce manuscrit, nous étudions la possibilité de répondre à ce problème de déséquilibre de charge dû à l'hétérogénéité et donc de perte d'efficacité en changeant de paradigme de parallélisme pour la programmation par tâches.

Chapitre 5

Programmation en tâches et support d'exécution

Ce chapitre se veut être une introduction générale sur le paradigme de parallélisme par tâches et les supports d'exécution servant à ordonnancer ces tâches. Nous présentons d'abord la notion de graphe de tâches et les différents modèles de programmation à base de tâches à partir d'un algorithme simple. Nous dressons ensuite un état de l'art des supports d'exécution pour la programmation en tâches ainsi que deux classifications, par modèles et par architectures. Nous terminons par une présentation du support d'exécution *PaRSEC*, celui sur lequel nous allons porter *DIVA*.

5.1 Paradigme de parallélisme par tâches

La programmation à base de tâches consiste à découper le travail en morceaux – appelés les tâches – et à décrire le flot des données entre ces morceaux (en anglais *dataflow*¹). Les dépendances entre ces tâches sont définies par les conditions de Bernstein [Ber66]. Soient deux tâches T_1 et T_2 . Soient (I_1, O_1) et (I_2, O_2) leurs couples de variables d'entrée et de sortie, respectivement. Il existe alors trois types de dépendance :

- si $O_1 \cap I_2 \neq \emptyset$, c'est une dépendance de donnée,
- si $O_2 \cap I_1 \neq \emptyset$, c'est une anti-dépendance (de donnée),
- si $O_1 \cap O_2 \neq \emptyset$, c'est une dépendance de sortie.

Les tâches sont indépendantes et peuvent être exécutées en parallèle si et seulement si ces trois ensembles sont vides. Paralléliser un code revient donc à déterminer les dépendances entre les tâches élémentaires qui le composent.

Dans la suite du manuscrit, les tâches font référence à des fonctions à exécuter au sein d'un unique programme. Pour autant, le paradigme de parallélisme par tâches ne se limite pas à ce niveau. Il est tout à fait possible de gérer l'exécution de plusieurs programmes, dont les tâches sont les programmes eux-mêmes. Par exemple la gestion de programmes sur une grille informatique (i.e. un ensemble de machines délocalisées, autonomes et très hétérogènes) est appelée *workflow* et l'intergiciel DIET [CD06] est un exemple de mise en œuvre de ce

1. Une architecture de processeurs est basée sur cette idée de flot des données, inventée dans les années 1970 par Jack Dennis [Den74]. Cependant, l'apparition du paradigme d'exécution dans le désordre (en anglais *Out of Order execution*) est devenu rapidement largement prédominant au début des années 1990.

principe, notamment sur la grille Grid5000 [BCC⁺06].

Nous allons maintenant construire le graphe de tâches d'un algorithme simple pour introduire les notions et définir deux modèles de programmation. Dans le chapitre suivant, nous appliquerons ce processus pour porter le code DIVA.

5.1.1 Du code séquentiel au graphe de tâches

Soit un pseudo-code basique réalisant l'algorithme 5.1, correspondant à l'exécution de quatre fonctions. L'ordre d'appel des fonctions *fun2* est ici arbitraire. En considérant chaque fonction comme une tâche, il est possible de représenter la réalisation du programme sous forme d'un graphe. Les tâches forment alors les nœuds et les dépendances, c'est-à-dire les flux de données, sont décrites par les arêtes (voir Fig. 5.1). Ces arêtes sont orientées et le graphe est acyclique, c'est ce que l'on appelle un DAG (*Directed Acyclic Graph*). Le programme, ainsi représenté dans un DAG, exhibe plus directement le parallélisme entre les tâches (ici sur *fun2*).

Algorithme 5.1 : Pseudo-code basique en séquentiel

Result : *C*

- 1 $A \leftarrow fun1();$
 - 2 $B \leftarrow fun2(A);$
 - 3 $C \leftarrow fun2(A);$
 - 4 $C \leftarrow fun3(B, C);$
-

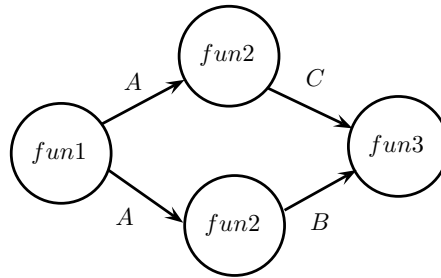


FIGURE 5.1 – DAG du pseudo-code basique

5.1.2 Différents modèles de programmation à base de tâches

Il existe plusieurs modèles de programmation à base de tâches. Ils sont similaires dans la mesure où ils décrivent le même DAG. Ils se distinguent dans la manière dont ils décrivent ce DAG. Nous présentons ici deux modèles bien que d'autres classifications pourraient être envisagées.

Le premier modèle de programmation à base de tâches est le *modèle séquentiel*, c'est celui qui se rapproche le plus de l'algorithme séquentiel 5.1. Les dépendances sont déterminées par rapport à l'ordre d'insertion des tâches. C'est l'exemple de l'algorithme 5.2, où les tâches sont définies par la primitive *InsertTask* qui consiste à insérer une tâche dans le DAG. La tâche

fun1 est insérée à la ligne 1. Cette insertion est une instruction non bloquante demandant au moteur d'exécution de traiter la tâche de manière asynchrone. Elle travaille avec la donnée *A* en écriture. Puis, la tâche *fun2* est insérée à la ligne 2. Elle travaille avec les données *A* en lecture et *B* en écriture. À partir de là, une dépendance de donnée sur *A* est identifiée entre les tâches *fun1* et *fun2*. Ainsi, le moteur d'exécution ne traitera la tâche *fun2* qu'une fois la tâche *fun1* terminée. Puis, ligne 3, la seconde tâche *fun2* est insérée. Elle travaille avec les données *A* en lecture et *C* en écriture. De même, une dépendance de donnée sur *A* est identifiée entre les tâches *fun1* et *fun2*. Les tâches *fun2* travaillent toutes les deux sur la donnée *A* mais en lecture, il n'y a donc pas de dépendance de sortie sur la donnée *A* entre ces tâches. Enfin, la tâche *fun3* est insérée à la ligne 4, elle travaille sur les données *B* en lecture et *C* en lecture/écriture. Il y a une dépendance de donnée sur *B* entre la première tâche *fun2* et la tâche *fun3* et une dépendance de donnée sur *C* entre la seconde tâche *fun2* et la tâche *fun3*.

Algorithme 5.2 : Pseudo-code basique en tâches, modèle séquentiel

Result : *C*

```

1 insertTask(fun1, A[Write]);
2 insertTask(fun2, A[Read], B[Write]);
3 insertTask(fun2, A[Read], C[Write]);
4 insertTask(fun3, B[Read], C[ReadWrite]);

```

Le second modèle de programmation à base de tâches est le *modèle paramétré*. Il consiste à décrire explicitement les tâches identifiées de l'algorithme 5.1 (*fun1*, *fun2* et *fun3*), éventuellement à l'aide de paramètres, ainsi que leurs dépendances et les données sur lesquelles portent ces dépendances. Ici, nous allons utiliser un paramètre *n* pour différencier les appels à *fun2*.

- La tâche *fun1* est décrite aux lignes 1 – 4. Dans le détail, la ligne 1 définit un nom arbitraire, ici *Task_fun1*. Les lignes 2 et 3 spécifient que la tâche travaille sur une donnée *A* en écriture avec une dépendance sur les tâches *Task_fun2*(1) et *Task_fun2*(2). Le sens de la flèche indique que la donnée *A* est un argument de sortie de *Task_fun1* et un argument d'entrée de *Task_fun2*(\star). La ligne 4 indique que la liste des données est terminée et que la suite concerne le code à exécuter pour réaliser cette tâche. La ligne 5 est ainsi simplement un appel à la fonction *fun1*.
- Les deux tâches *fun2* sont décrites dans une seule tâche aux lignes 7 – 13. Le nom arbitraire de cette tâche est *Task_fun2*, ligne 7, qui dépend d'un paramètre *n*. Nous nous servons de ce paramètre pour différencier les deux tâches *fun2*, qui vaut alors un ou deux, comme l'indique la ligne 8. Cette tâche travaille sur deux données *A* en lecture et *X* en écriture. La ligne 9 précise la dépendance de *A* qui est un argument d'entrée et qui provient de la réalisation de la tâche *Task_fun1* (par le sens de la flèche). La donnée *X*, qui est un argument de sortie, a un nom arbitraire, elle a une double dépendance de donnée avec la tâche *Task_fun3* pour laquelle elle est un argument d'entrée. La distinction se fait par le test de condition sur le paramètre *n*. Le nom de cette donnée pour la tâche *Task_fun3* est alors *B* ou *C*, lignes 10 et 11. La ligne 12 indique la fin de la liste des données et la ligne 13 est l'appel générique à la fonction *fun2*.
- La tâche *fun3* est décrite aux lignes 15 – 19. Elle s'appelle *Task_fun3*, ligne 15. Elle dépend de deux données *B* en lecture et *C* en lecture/écriture. Ces sont des arguments

d'entrée provenant des réalisations de la tâche paramétrée *Task_fun2* (définissant deux dépendances de données), lignes 16 et 17. La ligne 18 indique la fin de la liste des données et la ligne 19 est l'appel à la fonction *fun3*.

- Le DAG est finalement créé à la ligne 21.

À ce stade, chaque type de tâche est défini et, à partir d'une instance de tâches, il est possible de déterminer ses successeurs et ses prédécesseurs. Une fois la description de toutes les tâches effectuée, la primitive *dag.enqueue* commence la traversée du DAG. À noter que, malgré l'absence d'ordre séquentiel, le moteur d'exécution peut construire un ordre topologique. En particulier, il détecte que *fun1* n'a pas de prédécesseur et commence donc par exécuter la tâche correspondante.

Algorithme 5.3 : Pseudo-code basique en tâches, modèle paramétré

```
Result : C
1 Task_fun1()
2 WRITE A → A Task_fun2(1)
3           → A Task_fun2(2)
4 BODY
5 A=fun1()
6
7 Task_fun2(n)
8 n=1..2
9 READ  A ← A Task_fun1()
10 WRITE X → (n==1) ? B Task_fun3()
11           → (n==2) ? C Task_fun3()
12 BODY
13 X=fun2(A)
14
15 Task_fun3()
16 READ B ← B Task_fun2(1)
17 RW C   ← C Task_fun2(2)
18 BODY
19 C=fun3(B,C)
20
21 dag.enqueue(Task_★);
```

Le modèle séquentiel permet une productivité élevée puisqu'il est proche de l'algorithme d'origine, tandis que le modèle paramétré requiert d'exprimer plus explicitement les dépendances. Ce point est relativement complexe, comme le montre l'algorithme 5.3 par rapport à l'algorithme 5.2, qui décrivent pourtant tous deux le même DAG (celui de la Fig. 5.1). Le modèle paramétré offre cependant plus de flexibilité à l'exécution pour explorer le DAG du fait qu'il est entièrement connu à l'avance.

Un autre modèle de programmation à base de tâches est très répandu mais ne repose pas sur la notion de graphe, c'est le *modèle de boucles*. Lorsque le travail à paralléliser se

5.2. État de l'art des supports d'exécution pour la programmation à base de tâches

trouve à l'intérieur d'une boucle, ce modèle de programmation plus simple, également à base de tâches, consiste en un découpage statique ou dynamique de la boucle à exécuter. Selon les supports d'exécution, les dépendances de données peuvent ne pas être obligatoirement à spécifier et la cohérence des données n'est alors pas toujours assurée. Le programmeur doit alors veiller à ce que l'algorithme reste juste en détaillant quelles données sont à partager et quelles données sont locales aux tâches (données dites privées). C'est le cas d'OpenMP par exemple. À noter que OpenMP 4.0 supporte également la notion de tâche et propose une interface de programmation relativement similaire au modèle séquentiel discuté ci-dessus.

Malgré sa complexité, nous avons choisi de porter le code DIVA en tâches sur le modèle paramétré afin de pouvoir exploiter au mieux les propriétés du DAG. Il s'agit, à notre connaissance, du premier portage d'un code industriel sur ce modèle et en particulier sur le moteur d'exécution PaRSEC.

5.2 État de l'art des supports d'exécution pour la programmation à base de tâches

Le terme support d'exécution (en anglais *runtime*) désigne simplement un outil dont les programmes se servent pour être exécutés. Il en existe à plusieurs niveaux. Les systèmes d'exploitations multiprocesseurs, ou les langages de programmation interprétés sont par exemple supportés par un moteur d'exécution. Les implémentations de la norme MPI peuvent également définir un support d'exécution. Nous pouvons également citer les langages PGAS (*Partitioned Global Address Space*) qui sont spécialisées dans l'exécution de parallélisme de boucles sur des processeurs à mémoire distribuée mais qui ne se basent pas sur la notion de tâches. En principe, l'exécution d'un programme à base de tâches a vocation à être supportée par un moteur d'exécution et dans ce qui suit, nous nous focalisons sur les supports d'exécution pour cette programmation en tâches.

Le rôle des supports d'exécution pour la programmation en tâches est d'orchestrer la réalisation des tâches en respectant leurs dépendances. Ils se chargent de la répartition des tâches sur les ressources matérielles disponibles (potentiellement hétérogènes) tout en assurant la cohérence des données. Certains d'entre eux prennent en charge les transferts de données distantes, par exemple entre des CPUs ne partageant pas un espace de mémoire locale ou entre CPU et GPU, automatiquement ou par le biais de tâches explicites.

Pour être efficaces, ils se servent de politiques d'ordonnancement (statiques ou dynamiques), ce qui ne signifie pas que les choix soient optimaux. En effet, trouver le meilleur ordonnancement pour un graphe de tâches quelconque est un problème d'optimisation NP-complet au sens fort².

5.2.1 Petit historique des supports d'exécution

Dès l'apparition des premières machines parallèles, dans les années 1980, des supports d'exécution émergent pour améliorer l'équilibrage de charge, voir par exemple le livre de Sarkar [Sar89], basé sur des graphes de tâche statiques. La subdivision et le choix de répartition

2. Un problème est dit NP-complet si sa résolution nécessite un temps d'exécution super-polynomial en fonction de la taille des données. Le sens fort indique que ce temps ne change pas en modifiant l'algorithme (base,...).

du travail se font avant l'exécution du code, en fonction du nombre de ressources. À cette époque, l'architecture des processeurs était encore séquentielle (mono-cœur) et les machines parallèles étaient généralement à mémoire partagée.

La notion de vol de tâches (en anglais *work-stealing*) apparaît ensuite, ce qui rend les ordonnanceurs dynamiques. Concrètement, les tâches peuvent maintenant être ordonnancées à l'exécution, dès qu'une ressource est disponible. Le meilleur exemple historique est celui de Cilk [BJK⁺96], apparu dès 1994 et toujours développé actuellement sous le nom de Cilk++ [Lei10]. Il est basé sur la gestion du modèle séquentiel et les tâches sont décrites avec le mot clé *spawn*. Les supports d'exécution basés sur le modèle paramétré deviennent eux aussi dynamiques avec, notamment, l'apparition d'un formalisme de paramétrisation PTG (*Parametrized Task Graph*), voir [CJ99] et récemment [DBB⁺14], qui permet de décrire de façon plus compact le DAG.

Le cas d'OpenMP [DM98] est particulier. Comme nous l'avons dit plus haut, c'est un des supports d'exécution de tâches spécialisés dans le modèle de boucles. Il repose entièrement sur des directives de compilation. Apparu en 1997, il ajoute le mot-clef de tâche (`task`) en 2008 dans sa version 3 mais sans notion de dépendances de données. Plus récemment, la version 4 tend vers un modèle séquentiel en ajoutant la possibilité de décrire des dépendances de données. Il est sûrement le support d'exécution de tâches le plus répandu aujourd'hui.

Avec l'architecture de processeurs multi-cœurs du début des années 2000, plusieurs ressources se retrouvent au sein d'un seul processeur, ce qui permet d'exprimer plus de parallélisme au niveau local. Citons par exemple, Intel TBB (*Threading Building Blocks*) [Rei07], qui a été développé dès 2006 par le constructeur Intel spécialement pour ses processeurs multi-cœurs, et SMPs [PBL07] en 2007 qui a maintenant évolué vers OMPs (voir plus bas).

La gestion de ressources distantes, en mémoire distribuée, se fait alors généralement par le programmeur. C'est le paradigme de parallélisme par passage de message qui est le plus utilisé, notamment par la norme MPI. Il existe tout de même des supports d'exécution qui prennent en charge ces transferts de mémoires distants, comme par exemple KAAPI en 2006 [GBP07] (son développement n'est plus poursuivi actuellement, au bénéfice de X-KAAPI, voir plus bas).

Plus récemment, à la fin des années 2000, la révolution des accélérateurs entraîne l'apparition de supports d'exécution dédiés, avec la gestion des transferts mémoire CPU-GPU, comme StarPU [ATNW11, Aug11], PaRSEC [BBD⁺13] (anciennement DAGuE [BBD⁺12]), X-KAAPI [GLFR13] et STARSs (devenu OMPs [DPA⁺08] par fusion avec SMPs, ClusterSs, CellSs, ...), venus de différents horizons. Ces supports d'exécution adressent aujourd'hui toutes les architectures, c'est-à-dire mémoire partagée, mémoire distribuée et accélérateurs.

Pour revenir à l'adressage des accélérateurs, des supports d'exécution basés sur le parallélisme de boucles existent aussi dans ce domaine. Citons OpenACC [Ope11], qui provient d'une demande industrielle, et, plus récemment, OpenMP version 4.0, qui ajoute également cette possibilité. En ce qui concerne les co-processeurs Intel Xeon Phi sortis en 2012, comme leur architecture est basée sur des CPUs classiques à mémoire partagée, reposant sur le même jeu d'instructions x86, la plupart des supports d'exécution permettent leur gestion.

Pour finir ce tour d'horizon, il existe des supports d'exécution qui adressent les architectures à mémoires partagée et distribuée (sans rappeler les précédents qui gèrent en plus les

5.2. État de l’art des supports d’exécution pour la programmation à base de tâches

accélérateurs), comme CnC [BBC⁺10] (très récemment en version 1.0) et Taggre [RHAT13]. Ce dernier est en cours développement à Total pour traiter un problème spécifique de granularité des tâches trop fine en faisant de l’agrégation de tâches. Les communications distantes y sont instanciées soit automatiquement (PaRSEC, StarPU, CnC), soit explicitement par des tâches (X-KAAPI, OMPSs, Taggre).

5.2.2 Classification et utilisation par la communauté

Les supports d’exécution cités plus haut appartiennent donc aux trois catégories : parallélisme de boucles, modèle paramétré et modèle séquentiel. Le tableau 5.1 résume cette répartition en précisant éventuellement une gestion de plusieurs modèles. Nous utilisons les signes “++” et “+” pour préciser lorsqu’un modèle est mis plus en avant dans les spécification ou les publications. Le tableau 5.2 décrit les architectures adressées par ces supports d’exécution, en précisant pour la gestion de la mémoire distribuée si elle est implicite *via* un gestionnaire de mémoire ou si elle doit être explicitée par le programmeur.

Il existe d’autres classifications possibles, comme le fait d’utiliser une API (*Application Programming Interface*) ou de fournir un compilateur *source to source*, ou encore le fait de gérer les conflits et copies de données. Ce dernier point est important, lorsqu’il n’est pas pris en charge c’est au programmeur à y veiller, par exemple en précisant si les données sont partagées ou privées, comme nous l’avons vu précédemment. Heureusement, cela concerne principalement les supports d’exécution pour le modèle de boucles, généralement pour des architectures à mémoire partagée.

<i>runtime</i>	modèle de boucles	modèle paramétré	modèle séquentiel
Cilk			++
Cilk++	++		++
OpenMP ≤ 3.1	++		
OpenMP 4.0	++		+
OpenACC	++		+
Intel TBB	+		++
OMPSs	+		++
Intel CnC	+	++	
PaRSEC		++	
StarPU			++
Taggre	+	++	
X-KAAPI	+		++

TABLE 5.1 – Classification des supports d’exécution en fonction du modèle de programmation en tâches

Les exemples d’utilisation de supports d’exécution basés sur des graphes de tâches se multiplient depuis quelques années. La communauté d’algèbre linéaire dense est la première à avoir intensivement utilisé ces outils. En effet, les tâches concernent alors des blocs d’opérations dont la taille et les dépendances sont entièrement déterminées au préalable, ce qui facilite l’optimisation de leur ordonnancement. Citons par exemple [BLKD08, QOQOC⁺08] pour les architectures à mémoire partagée, [BBD⁺11a] pour les architectures à mémoire distribuée et [QOIQOvdG09, AAD⁺10, AAD⁺11b, AAD⁺11a] pour l’adressage des accélérateurs.

<i>runtime</i>	mémoire partagée	mémoire distribuée	accélérateurs GPU
Cilk	X		
Cilk++	X		
OpenMP ≤ 3.1	X		
OpenMP 4.0	X		X
OpenACC	X		X
Intel TBB	X		
OMPSs	X	explicite	X
Intel CnC	X	implicite	
PaRSEC	X	implicite	
StarPU	X	implicite/explicite	X
Taggre	X	explicite	
X-KAAPI	X	explicite	X

TABLE 5.2 – Classification des supports d'exécution en fonction des architectures supportées

Ces succès ont conduit au développement de bibliothèques de solveurs denses comme, par exemple pour les architectures à mémoire partagée, PLASMA [KLY⁺13], basé sur le support d'exécution QUARK [YKD11], et FLAME [ZCvdG⁺09], basé sur SuperMatrix [CQQv07]; pour les architectures à mémoire distribuée, DPLASMA [BBD⁺11b], basé sur PaRSEC; et pour l'adressage des accélérateurs, MAGMA [DDG⁺12], étendu pour utiliser les supports d'exécution QUARK et StarPU.

Plus récemment, c'est l'algèbre linéaire creuse qui se voit améliorée par la programmation à base de tâches. Le travail sur l'équilibrage de charge y est plus délicat puisque les tâches ne sont plus parfaitement déterminées au préalable (taille et dépendances). Citons par exemple les méthodes directes [ABGL13, LFR⁺14], les méthodes itératives (Krylov) [AGG⁺12] ou les "algorithmes rapides" [LSAT13, ABC⁺14, Kri14, LY14].

5.3 Le support d'exécution PaRSEC

PaRSEC est développé à l'ICL (*Innovative Computing Laboratory, University of Tennessee, USA*) par une équipe supervisée par George BOSILCA. C'est un des rares supports d'exécution instanciant de façon automatique les communications distantes. Il est aussi capable de gérer les configurations ccNUMA³ (*cache-coherent Non-Uniform Memory Access*), ainsi que les accélérateurs (GPU et Intel Xeon Phi).

La bibliothèque de solveurs denses DPLASMA (version "mémoire distribuée" de PLASMA) est basée sur PaRSEC. Il est composé principalement d'un ordonnanceur dynamique et d'un compilateur *source-to-source*, pour interpréter la paramétrisation du graphe de tâches. Ceci est complété par un ensemble d'outils de débogage et d'analyse de performance, comme un gestionnaire de traces, pour suivre l'activité précise de chaque ressource, ou un générateur graphique pour visualiser le graphe de tâches.

Le modèle de description des tâches dans PaRSEC est paramétré. Nous avons vu plus haut que le PTG (*Parametrized Task Graph*) s'était démocratisé pour leur représentation. PaRSEC se sert d'une extension propre, appelée JDF (*Job Data Flow*), comme langage de description

3. Une configuration ccNUMA implique un coût d'accès variable à la mémoire partagée (voir section 6.2.3).

de la paramétrisation. Cela permet de se rapprocher du langage mathématique de l'algorithme, en utilisant une représentation symbolique, comme nous l'avons vu précédemment à l'algorithme 5.3 qui est donc un exemple de JDF.

Les tâches sont donc décrites par le flux de leurs données. Il s'agit de lister les données nécessaires au déclenchement d'une tâche, en prenant soin de préciser de quelles autres tâches ces données proviennent et vers quelles autres tâches elles se dirigent. Lors de l'exécution, PaRSEC ne construit pas l'ensemble du graphe mais se représente à chaque instant les tâches en cours ainsi que les tâches à suivre directement, ce qui correspond à une coupe du DAG.

Chaque nœud peut contenir un certain nombre de ressources parallèles accédant à une mémoire locale (au nœud) partagée. Les transferts de données au sein d'un nœud ne nécessitent pas de surcoût de traitement lourd par le *runtime*, ce qui ne signifie pas que l'accès à ces données n'ait pas de surcoût propre, comme par exemple dans une machine ccNUMA.

Pour gérer cela, PaRSEC permet d'agréger virtuellement des ressources partageant un même niveau de mémoire au sein d'une seule entité, un *virtual process*. Ainsi, l'ordonnement des tâches dans cette entité est optimisé par rapport à la localité des données. Par ailleurs, il est toujours possible de faire du vol de tâches entre les *virtual processes*, pour corriger un éventuel déséquilibre de charge.

Nous avons choisi d'utiliser PaRSEC comme support d'exécution pour la programmation du code DIVA en tâches parce qu'il permet d'adresser toutes les architectures et qu'il prend en charge de façon automatique les communications distantes. En effet, nous souhaitons obtenir une bonne efficacité sur des machines diverses comme ccNUMA ou Intel Xeon Phi. De plus, nous verrons qu'une grande partie du code peut être réutilisée pour passer d'une architecture à une autre (le moteur d'exécution prenant en charge la gestion des détails techniques architecturaux), ce qui permet de se concentrer sur les besoins algorithmiques dans le but de maintenir une performance élevée. Par contre PaRSEC ne permettait pas d'utiliser un code en Fortran facilement et nous avons profité de ce projet en collaboration pour ajouter cette fonctionnalité au support d'exécution.

Chapitre 6

Formulation à base de tâches du code de propagation des ondes

Ce chapitre détaille les différentes étapes pour passer au paradigme de parallélisme à base de tâches dans le code DIVA. Nous commençons par créer le DAG associé, à partir de l’algorithme du pseudo-code parallèle présenté au chapitre 4. Une fois les tâches identifiées nous les décrivons suivant le modèle paramétré et plus précisément en JDF, le langage de description de tâches de PaRSEC. Nous allons ensuite montrer comment exploiter efficacement les différentes architectures cibles. Pour cela, il faut diminuer la granularité des tâches en générant plus de sous-domaines qui seront alors plus petits, tout en conservant une certaine localité. Il devient alors possible d’exploiter le vol de tâches. Nous illustrons ici l’importance de ces deux points clés sur un processeur classique. Nous terminons en décrivant la configuration ccNUMA et l’option de PaRSEC qui permet de la gérer efficacement.

6.1 DAG et description JDF des tâches du code DIVA

6.1.1 DAG du code DIVA

Nous avons vu au chapitre 4 que le code DIVA fait intervenir deux variables : le vecteur vitesse et le tenseur des contraintes. Nous utiliserons les notations V et S respectivement pour nous référer à chacune d’elles. L’algorithme parallèle 4.2 repose sur une décomposition de domaine initiale et une boucle en temps dans laquelle se situent deux boucles en espaces de mise à jour des variables ainsi que deux phases de communication, au début de chaque boucle. C’est un algorithme du type “*compute and exchange*”, sur deux variables à chaque pas de temps. Chaque domaine possède ses propres variables.

Voici comment en dériver le DAG correspondant. Nous appelons les fonctions de mise à jour *Compute_V* et *Compute_S*, il y en a une par sous-domaine et par pas de temps. Les phases de communication vont être instanciées automatiquement par le support d’exécution mais il faut néanmoins préparer les données à échanger. Pour cela, nous utilisons des tâches de préparation et de répartition, il y en a autant par sous-domaine que le nombre de sous-domaines voisins, pour chaque pas de temps. Les premières, *Pack_V* et *Pack_S*, se chargent de remplir un vecteur avec les données à envoyer (situés sur les degrés de liberté des faces externes au sous-domaine). Les secondes, *Unpack_V* et *Unpack_S*, mettent à jour les variables principales avec les données reçues.

Chaque itération en temps fait ainsi appel aux mêmes tâches : *Compute_V*, *Compute_S*, *Pack_V*, *Pack_S*, *Unpack_V*, *Unpack_S* avec les mêmes dépendances de données. Nous avons représenté sur la Fig. 6.1 le DAG de DIVA pour une itération et pour trois sous-domaines comme décrits à la Fig. 4.2. Les dépendances sont représentées par les flèches avec le nom de la donnée concernée. Il n’y a pas de notion d’envoi ou de réception avec d’autres sous-domaines. Au moment de l’exécution, un processeur peut très bien avoir plusieurs sous-domaines à gérer ou plusieurs processeurs peuvent avoir leurs sous-domaines situés sur un espace de mémoire partagée. Dans ce cas, une copie de donnée locale suffit et c’est le gestionnaire de mémoire du support d’exécution qui se charge de ces copies ou si besoin des envois et des réceptions.

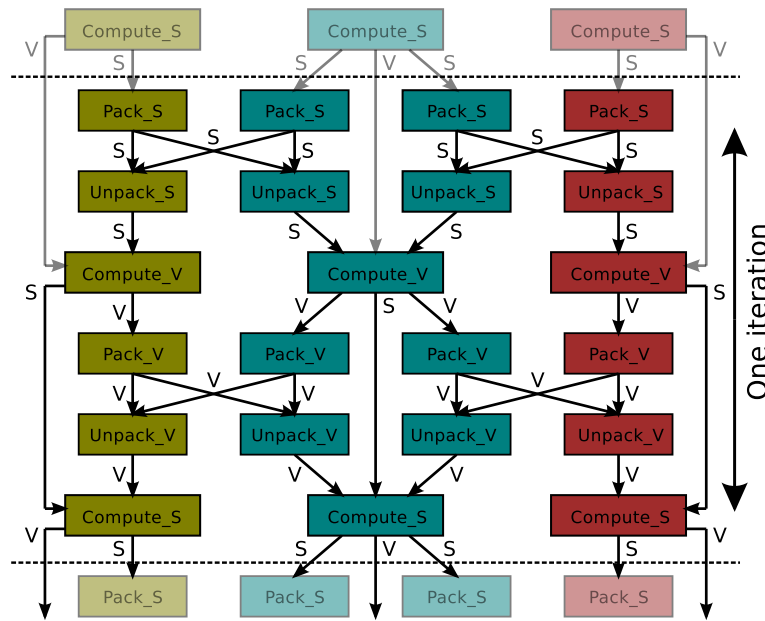


FIGURE 6.1 – DAG correspondant à une itération temporelle de l’algorithme DIVA pour un maillage découpé en trois sous-domaines comme dans la Fig. 4.2

6.1.2 Description paramétrée en JDF

La description du DAG de DIVA peut se faire de différentes manières en fonction du support d’exécution, voir chapitre 5. Nous avons vu que PaRSEC se sert d’un langage propre pour cela, le JDF (*Job Data Flow*). Nous allons donc utiliser le modèle paramétré présenté à la section 5.1 pour décrire les tâches du DAG 6.1.

Comme nous l’avons dit plus haut, chaque sous-domaine fait appel aux mêmes tâches et ce pour chaque itération temporelle. Nous allons donc paramétrer les tâches avec

- $it = 1..Nt$, l’itération courante, comprise entre 1 et Nt , le nombre total d’itérations,
- $d = 1..Nd$, le sous-domaine local, compris entre 1 et Nd , le nombre total de sous-domaines.

De plus, les tâches de communication seront paramétrées avec $d_n = 1..nb_neigh(d)$, la liste des voisins d’un domaine d , variant de 1 à Nd , le nombre de voisins du domaine d . Pour cela,

nous avons ajouté une fonction *nb_neigh()* qui renvoie le nombre de voisins d'un sous-domaine.

Nous allons décrire les tâches sur la variable vitesse *V* seulement (algorithmes 6.1, 6.2 et 6.3), les tâches sur la variable contraintes *S* possédant des structures tout à fait symétriques (algorithmes 6.4, 6.5 et 6.6 respectivement).

- L'algorithme 6.1 décrit la tâche *Compute_V*, ligne 1, pour la mise à jour de *V*. Les paramètres de l'itération courante et du sous-domaine courant sont listés aux lignes 2 et 3 respectivement. Cette tâche a deux variables *S* et *V*, la première en lecture et la seconde en lecture/écriture, comme cela est précisé aux lignes 4 et 6. La variable *S* provient des tâches *Unpack_S* de l'itération courante, du sous-domaine courant et de chaque voisin. Cette information est incluse ligne 4 dans les paramètres de $(it, d, 1..nb_neigh(d))$. Cette variable servira ensuite pour la tâche *Compute_S* de l'itération courante et du sous-domaine local, ligne 5. La seconde variable *V* provient de la tâche *Compute_S* de l'itération précédente et du sous-domaine local, ligne 5, paramètres $(it - 1, d)$. La distinction si ce n'est pas la première itération se fait par le test $[(it \neq 1) ?]$. Cette variable est ensuite nécessaire aux tâches *Pack_V* de l'itération courante, du sous-domaine courant et de chaque indice de voisin, ligne 7. La description de la tâche se termine par la section BODY, ligne 8, qui contient l'appel à la fonction de mise à jour *V*, ligne 9.
- L'algorithme 6.2 décrit la tâche *Pack_V*, ligne 1, pour la préparation des données de *V* à envoyer. Les paramètres de l'itération courante, du sous-domaine courant et de l'indice du voisin considéré sont listés aux lignes 2, 3 et 4 respectivement. Cette tâche a une seule variable *V* en lecture, ligne 5. Elle provient de la tâche *Compute_V* de l'itération courante et du sous-domaine courant, ligne 5. Cela renvoie à la ligne 7 de l'algorithme 6.1. Cette variable servira ensuite pour la tâche *Unpack_V* de l'itération courante, du sous-domaine courant et faisant référence au même indice de voisin, ligne 6. Dans ce cas elle servira à recevoir les données d'un voisin distant. Mais cette variable doit également être envoyée à son voisin direct. Pour cela, la ligne 7 précise que la variable est nécessaire à une autre tâche *Unpack_V* de l'itération courante, dont les deux derniers paramètres se basent sur des appels de fonctions faisant correspondre un sous-domaine à ses voisins et fournissant leurs indices dans la numérotation distante : *num_dom()* et *index_at_neigh()* respectivement. La description de la tâche se termine par la section BODY, ligne 8, qui contient l'appel à la fonction de préparation sur *V*, ligne 9.
- L'algorithme 6.3 décrit la tâche *Unpack_V*, ligne 1, pour la répartition des données de *V* reçues. Les paramètres de l'itération courante, du sous-domaine courant et de l'indice du voisin considéré sont listés aux lignes 2, 3 et 4 respectivement. Cette tâche a une seule variable *V* en lecture/écriture, ligne 5. Elle provient de la tâche *Pack_V* de l'itération courante, du sous-domaine courant et faisant référence au même indice de voisin, ligne 5. Cela renvoie à la ligne 6 de l'algorithme 6.2. Elle provient également de la tâche *Pack_V* de l'itération courante, du sous-domaine voisin et de l'indice distant, tous deux déterminés à partir des fonctions *num_dom()* et *index_at_neigh()*, ligne 6. Cela renvoie à la ligne 7 de l'algorithme 6.2. Cette variable servira ensuite à la tâche *Compute_S* de l'itération courante et du sous-domaine courant, ligne 7. La description de la tâche se termine par la section BODY, ligne 8, qui contient l'appel à la fonction de répartition sur *V*, ligne 9.

Nous avons ainsi décrit tous les nœuds du DAG en fonction de paramètres. Pour chaque

Algorithme 6.1 : Description JDF de la tâche *Compute_V*

```

1 Compute_V(it, d)
2   it = 1..Nt
3   d = 1..Nd
4 READ S ← S Unpack_S(it, d, 1..nb_neigh(d))
5       → S Compute_S(it, d)
6 RW V ← (it ≠ 1) ? V Compute_S(it - 1, d)
7       → V Pack_V(it, d, 1..nb_neigh(d))
8 BODY
9 call_compute_V(S, V);

```

Algorithme 6.2 : Description JDF de la tâche *Pack_V*

```

1 Pack_V(it, d, dn)
2   it = 1..Nt
3   d = 1..Nd
4   dn = 1..nb_neigh(d)
5 READ V ← V Compute_V(it, d)
6       → V Unpack_V(it, d, dn)
7       → V Unpack_V(it, num_dom(d, dn), index_at_neigh(d, dn))
8 BODY
9 call_pack_V(V);

```

Algorithme 6.3 : Description JDF de la tâche *Unpack_V*

```

1 Unpack_V(it, d, dn)
2   it = 1..Nt
3   d = 1..Nd
4   dn = 1..nb_neigh(d)
5 RW V ← V Pack_V(it, d, dn)
6       ← V Pack_V(it, num_dom(d, dn), index_at_neigh(d, dn))
7       → V Compute_S(it, d)
8 BODY
9 call_unpack_V(V);

```

donnée (les variables V et S), il est précisé de quelle tâche elle provient, éventuellement de l'itération précédente et pour quelle tâche elle sera nécessaire, éventuellement à l'itération suivante. L'ensemble de ces informations est équivalent à l'ensemble des arêtes du DAG. Les étapes de décomposition de domaines et d'initialisation des données ne sont pas incluses dans le graphe de tâche dans le cadre de ce travail.

À noter que, ainsi décrites, les arêtes encodent non seulement les dépendances entre les tâches mais également les données sur lesquelles elles portent. *PaRSEC* est donc en mesure d'effectuer automatiquement les transferts de données à partir de ces informations s'il a besoin de déplacer des données, comme c'est le cas sur une architecture à mémoire distribuée. Toutefois, dans le cadre de ce manuscrit, nous n'évaluons pas cette possibilité qui est en cours de développement.

Algorithme 6.4 : Description JDF de la tâche *Compute_S*

```

1 Compute_S(it, d)
2   it = 1..Nt
3   d = 1..Nd
4 READ V ← V Unpack_V(it, d, 1..nb_neigh(d))
5       → (it ≠ Nt) ? V Compute_V(it + 1, d)
6 RW S ← S Compute_V(it, d)
7       → (it ≠ Nt) ? S Pack_S(it + 1, d, 1..nb_neigh(d))
8 BODY
9 call_compute_S(V, S);
```

Algorithme 6.5 : Description JDF de la tâche *Pack_S*

```

1 Pack_S(it, d, dn)
2   it = 1..Nt
3   d = 1..Nd
4   dn = 1..nb_neigh(d)
5 READ S ← (it ≠ 1) ? S Compute_S(it - 1, d)
6       → S Unpack_S(it, d, dn)
7       → S Unpack_S(it, num_dom(d, dn), index_at_neigh(d, dn))
8 BODY
9 call_pack_S(S);
```

Algorithme 6.6 : Description JDF de la tâche *Unpack_S*

```

1 Unpack_S(it, d, dn)
2   it = 1..Nt
3   d = 1..Nd
4   dn = 1..nb_neigh(d)
5 RW S ← S Pack_S(it, d, dn)
6       ← S Pack_S(it, num_dom(d, dn), index_at_neigh(d, dn))
7       → S Compute_V(it, d)
8 BODY
9 call_unpack_S(S);
```

Il reste donc maintenant à appeler *PaRSEC* pour exécuter le DAG du code *DIVA*. Cependant, nous allons voir qu'il faut encore au moins modifier la granularité des tâches et activer le vol de tâches pour pouvoir corriger les problèmes d'équilibrage de charge sur une architecture simple.

6.2 Extensions pour architectures à mémoire partagée

6.2.1 Granularité et localité

Dans notre modèle de programmation, le DAG est *a priori* décorrélé de l'architecture. En particulier, le nombre de sous-domaines n'est plus dépendant du nombre d'unités de calculs

disponibles, comme c'était le cas pour l'implémentation par MPI. Il est donc possible de faire varier la granularité des tâches en exprimant d'avantage de parallélisme. L'idée est de diminuer le grain en créant plus de sous-domaines, qui deviennent alors plus petits. La charge de travail de chaque tâche est ainsi réduite. Les étapes de communications impliquent alors moins de données mais sont plus nombreuses. Cependant, elles sont indépendantes puisqu'elles agissent sur des sous-domaines différents.

Plusieurs sous-domaines peuvent être affectés à une unité de calculs. Afin de favoriser la localité des données, le maillage global est d'abord découpé en un nombre de sous-domaines égal au nombre d'unités de calculs, puis, chacun de ces sous-domaines est subdivisé en fonction de la granularité choisie. La Fig. 6.2 illustre cela sur l'exemple précédent du maillage 2D à trois sous-domaines (voir Fig. 4.2). Chaque sous-domaine en trait continu est subdivisé en quatre, en traits discontinus, permettant un découpage du domaine initial en 12 sous-domaines avec une localité pour chaque groupe de sous-domaines (en couleurs) qui sont alors adjacents. Les données à échanger entre ces sous-domaines se feront alors par simples copies locales.

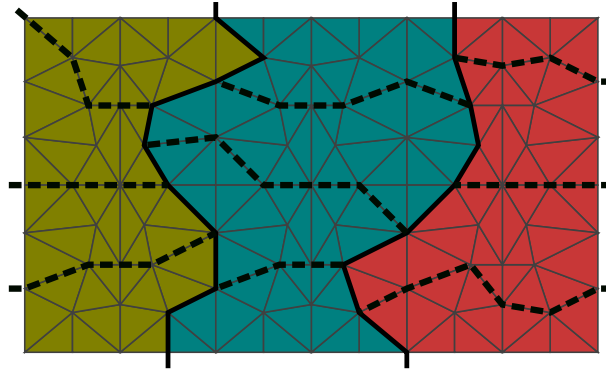


FIGURE 6.2 – Exemple de subdivision dans un maillage 2D

Cette possibilité de gestion du nombre de sous-domaines indépendants du nombre d'unités de calculs a été ajoutée au code DIVA dont la décomposition de domaine initiale était fixée au nombre de processus MPI. La fonction de subdivision étant appelée sur chaque sous-domaine, il est possible de créer un nombre quelconque de niveaux de décomposition par récursivité. Cela est notamment nécessaire pour une meilleure gestion des architectures à mémoire distribuée des nœuds contenant des coprocesseurs.

Il n'est pas nécessaire de modifier la description des tâches. Par contre, il faut préciser à l'initialisation de *PaRSEC* la localité des sous-domaines à traiter de préférence sur chaque unités de calcul. Le vol de tâches permet alors à une unité de calcul de déclencher des tâches sur des sous-domaines qui ne font pas partie de ses préférences mais qui se situent sur le même espace de mémoire partagée. Une fois le vol effectué, les préférences des unités de calculs ne sont pas modifiées.

6.2.2 Illustration expérimentale

Nous nous concentrons pour cette expérience sur l'illustration des points-clés d'un support d'exécution pour la programmation à base de tâches, à savoir le vol de tâches et l'importance

6.2. Extensions pour architectures à mémoire partagée

de la granularité fine. C'est aussi l'occasion d'observer que, même sur un seul processeur, la programmation en tâches peut être bénéfique.

Le cas test considéré est le cube 3D hétérogène dont les spécificités sont détaillées à la section 7.1. Nous considérons deux tailles de maillages constitués de 200.000 et 800.000 tétraèdres (générés avec Tetgen). La machine est ici formée d'un seul processeur classique, Intel Xeon E7-8837, composé de 8 CPUs cadencés à 2.67 GHz. Ils possèdent chacun 64 KB et 256 KB de cache L1 et L2 respectivement, et se partagent 24 MB de cache L3.

La Fig. 6.3 représente le *speedup* obtenu avec le code DIVA en tâches exécutées avec PaRSEC, par rapport au code initial en MPI. Nous rappelons que la granularité du code MPI correspond au découpage du domaine en autant de sous-domaines que d'unités de calculs disponibles, ici de 1 à 8 CPUs. Nous observons que, premièrement, il apparaît clairement qu'utiliser le support d'exécution avec la même granularité (grossière) que MPI n'apporte aucun gain (barres rouges traits pleins). En diminuant cette granularité, c'est-à-dire en découpant le domaine en plus de sous-domaines (plus petits), les communications de données se font sur des interfaces plus petites, certes plus nombreuses aussi, mais de façon indépendantes. Le fait d'exprimer plus de parallélisme permet alors un meilleur ordonnancement, observé sur les autres barres. Nous observons également que, sans activer le vol de tâches (barres vertes traits discontinus), le gain reste faible et le mauvais équilibrage de charge est prédominant. En activant le vol de tâches (barres bleues traits pointillés), le gain est plus significatif. Dès qu'une ressource est disponible, le support d'exécution a plus de choix sur les tâches à activer pour optimiser les temps d'attente dus aux communications. Sur un seul CPU du processeur, il n'y a évidemment aucune différence à activer ou non le vol de tâches.

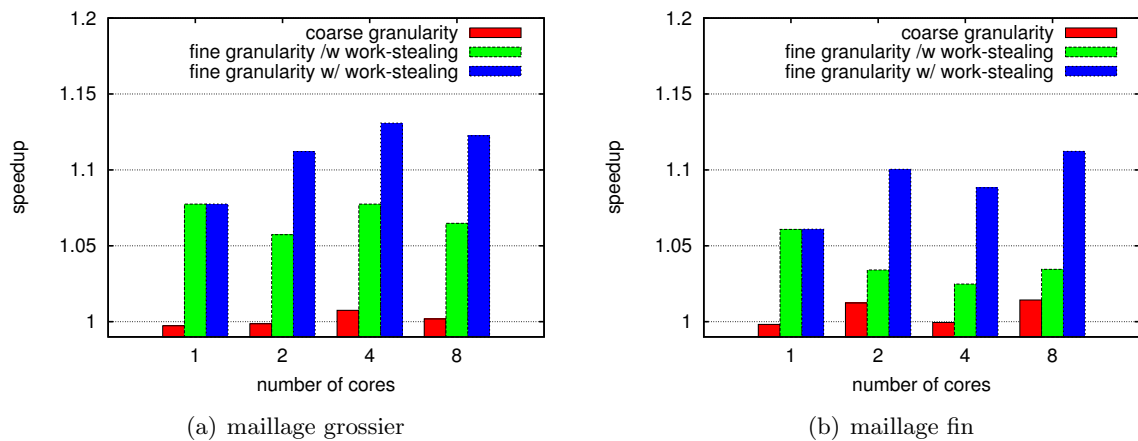


FIGURE 6.3 – *Speedup* sur un processeur, effets de la granularité et du vol de tâches

Cette expérience démontre qu'il est possible d'avoir un gain même sur peu de CPU et que ce gain est maximal si la granularité est fine et si le vol de tâches est activé.

6.2.3 Gestion des nœuds ccNUMA

Le terme ccNUMA, de l'anglais *cache coherent Non-Uniform Memory Access*, désigne une architecture de machine parallèle à mémoire partagée. Dans une configuration NUMA, la mémoire est physiquement répartie en bancs (ou nœud, mais ce terme peut être confondu avec un nœud de processeurs). Chaque banc est localisé près d'un processeur, lui permettant un accès plus rapide. À l'inverse, les autres processeurs ont un temps d'accès à ce banc plus élevé, dépendant de la distance physique les séparant. C'est le facteur NUMA.

Lorsque les processeurs disposent de mémoire cache et que le système assure la cohérence des données entre la mémoire globale (bancs) et les caches internes, la configuration est précisée *cache coherent* et abrégée ccNUMA. La Fig. 6.4 représente un nœud d'architecture ccNUMA avec 4 processeurs. Le plus souvent, les processeurs sont groupés par deux, modifiant légèrement le facteur NUMA.

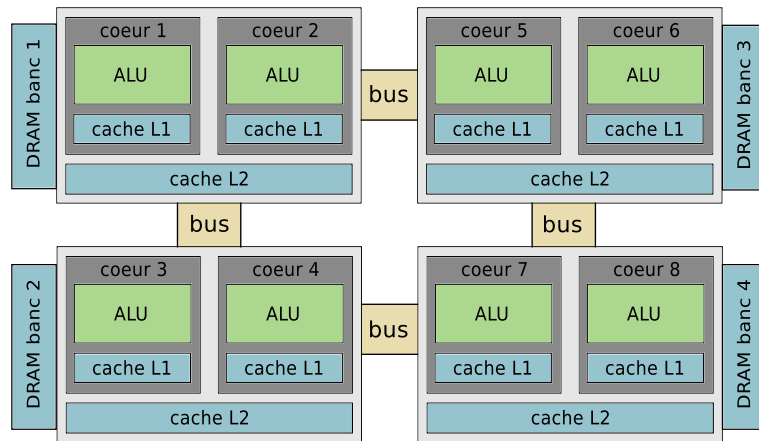


FIGURE 6.4 – Exemple d'architecture ccNUMA à 4 processeurs

PaRSEC permet de gérer cette mémoire partagée à coûts d'accès variables en utilisant ses *virtual processes* (VP), décrits à la section 5.3. Pour cela, il n'est pas nécessaire de modifier la description des tâches. Une option à l'exécution permet de définir le nombre d'unité de calculs à agréger virtuellement. Par défaut l'option définit un seul espace de mémoire partagée et il est possible de l'utiliser de deux façons :

- en précisant manuellement un découpage régulier : $-V \text{ } rr : \{N\} : \{S\} : \{T\}$ où N est le nombre de VP, S est le nombre d'unités de calcul de chaque VP et T est le nombre total d'unités de calcul ($T = N * S$)
- en utilisant l'outil *hwloc* [BCOM⁺10] : $-V \text{ } hwloc$ qui détecte automatiquement les ressources disponibles et qui déduit de leurs architectures le nombre de VP et le nombre d'unités de calcul dans chacun d'eux.

Chapitre 7

Étude expérimentale

Ce dernier chapitre illustre les résultats obtenus sur plusieurs machines cibles avec le code DIVA programmé en tâches et ordonnancé par PaRSEC. Ils sont comparés à l'implémentation initiale utilisant le paradigme de parallélisme par passage de message de la bibliothèque MPI. Nous commençons par présenter le cas test physique, un cube 3D contenant un dôme de sel. Nous listons ensuite les différentes configurations matérielles que nous utilisons, ccNUMA et Intel Xeon Phi. Cette étude ne concerne donc que des architectures à mémoire partagée. Nous montrons que les résultats de comparaison de *speedup* donnent l'avantage à la version en tâches avec PaRSEC et que les efficacités sont toujours meilleures que la version MPI.

7.1 Description des cas tests

Géophysique d'un dôme de sel 3D

L'ensemble de l'étude expérimentale est réalisé sur le même cas test géophysique. Pour se rapprocher des cas de prospection pétrolière utilisés en RTM, nous avons construit un domaine artificiel 3D. Il représente un dôme de sel (en blanc) sous plusieurs couches géologiques, elles-mêmes sous une couche d'eau (en bleu), comme illustré Fig. 7.1. Chaque couche possède des caractéristiques physiques différentes, avec des extrema dans les couches d'eau et de sel.

Deux discrétisations hétérogènes sont proposées, formant des maillages non-structurés de tétraèdres. Le plus grossier est composé de 200.000 mailles, le plus fin de 800.000, tous deux générés avec le logiciel Tetgen. Le nombre de degrés de liberté (noté `nb_ddl`) de chaque maille varie de 4 à 20 en fonction des caractéristiques de chaque milieu, voir tableau 7.1.

couche	ρ	V_p	V_s	ε	δ	γ	θ	ϕ	nb_ ddl
bleue	1	1500	750	0	0	0	0	0	4
jaune	1	1800	900	0.1	0.01	0.1	0	0	10
brune	1	2000	1000	0.15	0.05	0.15	15	30	10
noire	1	2500	1250	0.25	0.01	0.2	15	30	20
blanche	1	4000	2000	0	0	0	0	0	20

TABLE 7.1 – Caractéristiques des couches géologiques du cas test, en m.s^{-1} pour les vitesses

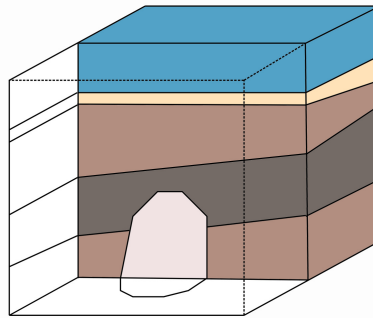


FIGURE 7.1 – Cas test géologique d'un dôme de sel 3D

Architectures à mémoire partagée

Le banc d'essai de l'étude expérimentale sur architectures à mémoire partagée porte sur trois configurations machines, deux nœuds ccNUMA et un coprocesseur :

- Un nœud classique formé de 2 processeurs Intel Xeon E5-2670 v2, composé chacun de 10 CPUs cadencés à 2.50 GHz. Chaque cœur CPU possède 64 KB et 256 KB de cache L1 et L2 respectivement, et chaque processeur dispose de 25 MB de cache L3. Cette configuration est dénommée par la suite *20-core Xeon*.
- Un nœud plus atypique, formé de 8 processeurs Intel Xeon E7-8837, composé chacun de 8 CPUs cadencés à 2.67 GHz. Chaque cœur CPU possède 64 KB et 256 KB de cache L1 et L2 respectivement, et chaque processeur dispose de 24 MB de cache L3. Cette configuration est dénommée par la suite *64-core Xeon*.
- Un coprocesseur *many-cœurs* Intel Xeon Phi 7120P, composé de 61 CPUs cadencés à 1.238 GHz, avec un hyperthreading par 4, totalisant 244 threads. Chaque cœur CPU possède 64 KB et 512 KB de cache L1 et L2 respectivement. Il n'y a pas de cache L3. Cette configuration est dénommée par la suite *Xeon Phi*.

Facteurs NUMA des machines de tests

Le tableau 7.2 présente les distances entre les bancs mémoire de la machine *20-core Xeon*. Le nombre 10, correspondant à la distance d'accès au banc mémoire local est indicatif, il ne représente pas de coût réel. La distance d'accès au banc mémoire distant est à diviser par la distance de référence pour donner un facteur NUMA de 2. Ce qui indique que le coût d'accès au banc distant sera deux fois plus élevé que sur le banc local.

Le tableau 7.3 présente les distances entre les bancs mémoire de la machine *64-core Xeon*. La distance de référence est aussi notée 10. Le facteur NUMA d'accès aux bancs distants varie de 1.3 à 4.8. Il est facile d'identifier sur cette table les bancs regroupés au sein d'une même groupe. Le temps d'accès au banc distant d'un groupe n'est multiplié que par 1.3 tandis que les autres bancs sont au minimum à 4 de distance.

banc	0	1
0	10	20
1	20	10

TABLE 7.2 – Facteurs NUMA de la machine *20-core Xeon*

banc	0	1	2	3	4	5	6	7
0	10	13	40	40	40	40	48	48
1	13	10	40	40	40	40	48	48
2	40	40	10	13	48	48	40	40
3	40	40	13	10	48	48	40	40
4	48	48	10	13	40	40	40	40
5	48	48	13	10	40	40	40	40
6	48	48	40	40	40	40	10	13
7	48	48	40	40	40	40	13	10

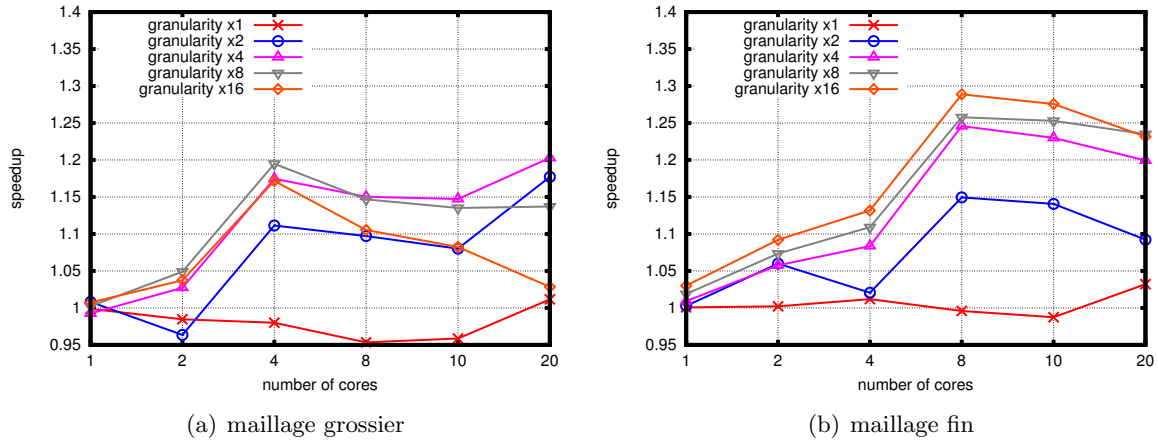
TABLE 7.3 – Facteurs NUMA de la machine 64-core Xeon

7.2 Résultats numériques

7.2.1 Machines à un nœud ccNUMA

Nœud classique 20-core Xeon

Les premiers tests concernent la configuration classique 20-core Xeon. La Fig. 7.2 présente les différents *speedup* obtenus sur les maillages grossier, 7.2(a) et fin, 7.2(b). Pour une granularité similaire au code MPI, en rouge trait plein, il n'y a pas de gain. Plus la granularité diminue, plus le gain est significatif, jusqu'à une certaine limite. Par exemple pour une granularité correspondant à 16 subdivisions de chaque sous-domaine initial, le *speedup* est moins bon que pour une granularité 8 et même 4 sur 20 cœurs CPU sur le maillage grossier.

FIGURE 7.2 – *Speedup* sur 20-core Xeon, pour différentes granularités

La Fig. 7.3 résume les résultats obtenus avec la meilleure granularité pour chaque expérience. Le *speedup* est présenté Fig. 7.3(a) et l'efficacité Fig. 7.3(b). Le code MPI a une efficacité décroissante au fur et à mesure que le nombre de cœurs CPU augmente. À l'inverse, le code utilisant PaRSEC produit une efficacité constante, très proche de 1, ce qui traduit une utilisation adéquate des ressources disponibles. Les courbes décrochent pour 20 cœurs CPU, ce qui est à imputer en partie au facteur NUMA de 2 qui intervient dans ce cas précis.

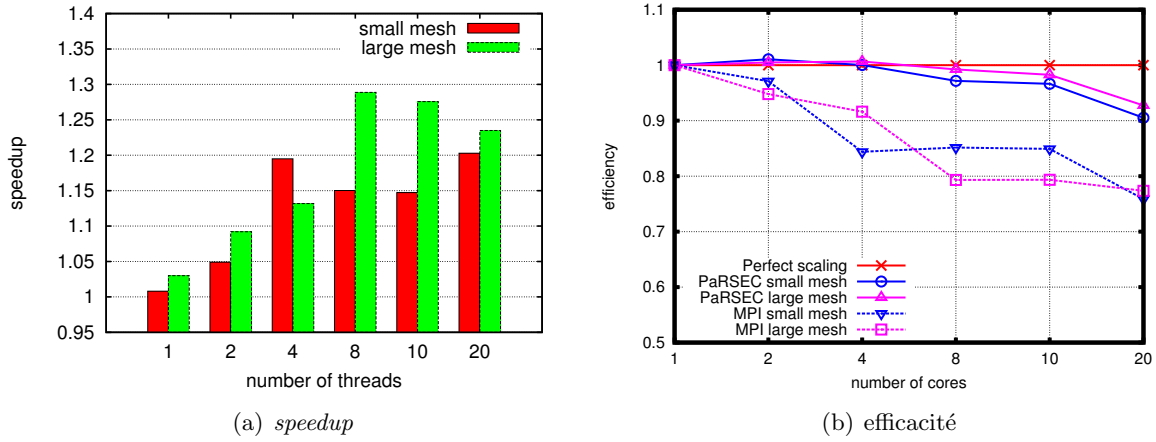


FIGURE 7.3 – Résultats 20-core Xeon, meilleur *speedup* et efficacité

Nœud atypique 64-core Xeon

Les tests concernent maintenant la configuration plus atypique 64-core Xeon. Nous rappelons que sur cette machine le facteur NUMA atteint un maximum de 4.8. L'utilisation des *virtual processes* (VP) de PaRSEC est alors incontournable. La Fig. 7.4 illustre cela sur un échantillon des résultats obtenus sur le maillage fin avec un nombre de processeurs variant de 16 à 48, soit un nombre de bancs NUMA variant de 2 à 6. La figure de gauche 7.4(a) ne différencie pas les bancs NUMA et considère la mémoire partagée à égale distance des cœurs. Cela conduit à des *speedup* inférieurs à 1, traduisant un code plus long que l'implémentation d'origine en MPI. En effet, l'option `numactl -localalloc` permet à une exécution MPI de ne pas être impactée par le facteur NUMA. La figure de droite 7.4(b) représente les résultats sur le même maillage fin, obtenus en utilisant un nombre de VP égal au nombre de bancs NUMA. Dans cette configuration, nous retrouvons le comportement observé sur la configuration 20-core Xeon, à savoir un *speedup* qui augmente lorsque la granularité des tâches diminue.

La Fig. 7.5 résume les résultats obtenus avec la meilleure granularité pour chaque expérience, sur les maillages fin et grossier. Le nombre de cœurs varie de 1 à 64, soit un nombre de bancs NUMA variant de 1 à 8. Les *speedup* du maillage fin (barres vertes traits pointillés) montrent un gain quel que soit le nombre de processeurs, ainsi qu'une efficacité excellente jusqu'à 32 cœurs et qui décroît légèrement à 64 cœurs. Pour le maillage grossier par contre, les *speedup* (barres rouges traits pleins) deviennent inférieurs à 1 à partir de 16 cœurs, ce qui se vérifie sur les courbes d'efficacité qui deviennent moins bonnes que l'implémentation de référence à partir de 16 cœurs. Cela s'explique par la taille du maillage qui produit des sous-domaines trop petits lorsque la granularité est fine, conduisant à un surcoût de traitement plus important que le temps de calcul effectif des tâches.

Ces résultats sont obtenus avec l'utilisation des *virtual processes* (VP) de PaRSEC. Chaque processeur, correspondant à 8 cœurs CPU, forme un VP, assurant une distinction entre le banc de mémoire local et les bancs distants. Le vol de tâche est ici limité au périmètre de chaque VP. (La Fig. 7.4 présente les résultats obtenus sur le maillage fin avec un nombre de processeurs variant de 16 à 48 en n'utilisant qu'un seul VP, c'est-à-dire en autorisant le vol de tâches sur

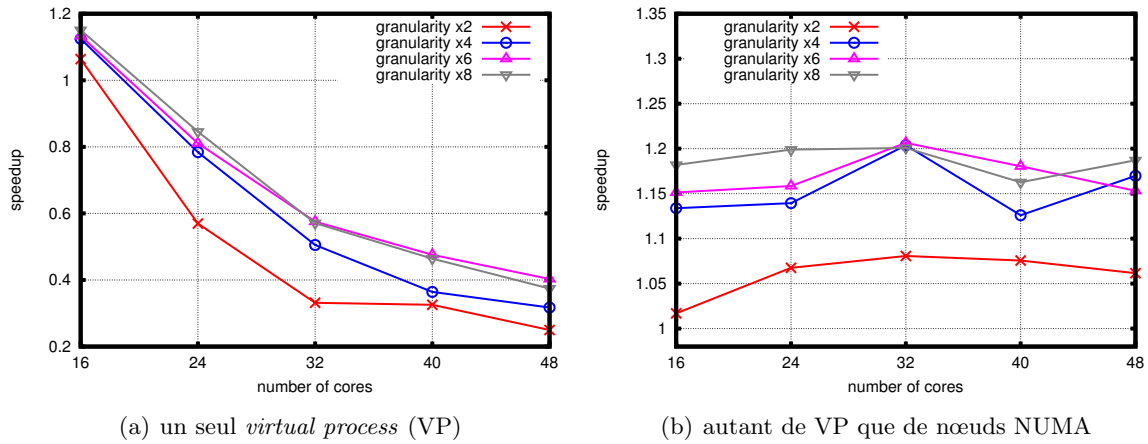


FIGURE 7.4 – Détails de résultats avec le maillage fin sur 64-core Xeon pour la prise en compte des facteurs NUMA

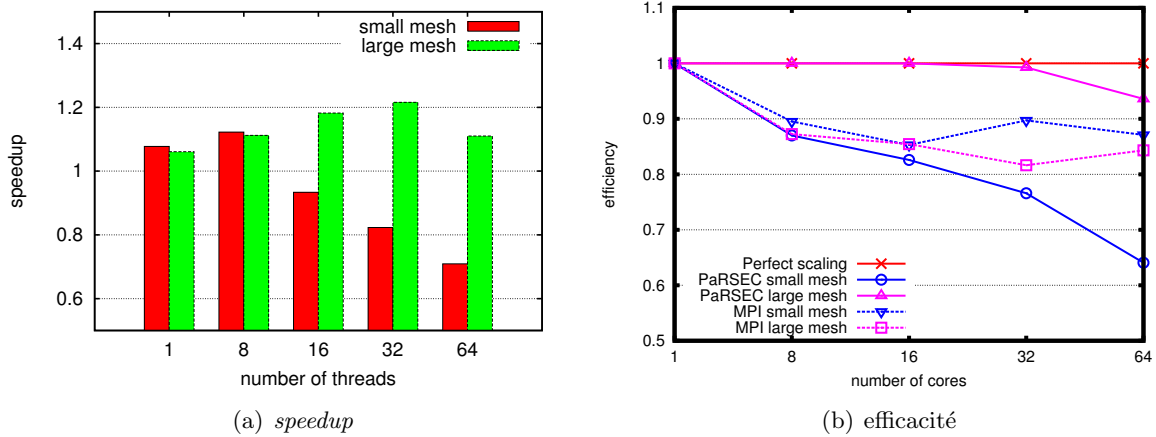


FIGURE 7.5 – Résultats 64-core Xeon, meilleur speedup et efficacité

tous les bancs mémoire. Cette fois les performances sont dans tous les cas moins bonnes que le code de référence, jusqu'à atteindre un *speedup* rédhibitoire de 0.3, soit un temps d'exécution multiplié par 3.) Le changement de granularité seul n'aide pas suffisamment pour combler les facteurs NUMA.

La Fig. 7.6 reprend l'exemple d'exécution sur 32 cœurs CPU du chapitre 4. Les traces représentent les temps d'activité et d'inactivité des ressources. Les tâches de calculs du vecteur vitesse et du tenseur des contraintes sont représentées en orange et rouge respectivement. Le gris clair met en évidence les temps d'attente, notamment dans le code MPI, Fig. 7.6(a), où la répartition de charge n'est pas parfaite. La Fig. 7.6(b) correspond à une granularité divisée par 6 par rapport au code MPI. Il n'y a presque plus de temps d'attente par rapport au cas MPI et la partie en gris foncé illustre le temps gagné.

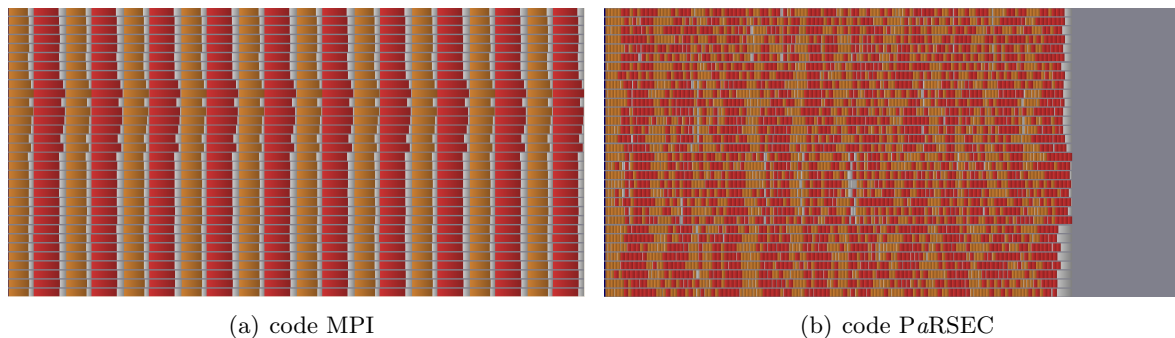


FIGURE 7.6 – Comparaison de traces d’exécution sur 32 cœurs, MPI et PaRSEC, à la même échelle de temps (la partie gris foncé de la figure (b) correspond au gain)

7.2.2 Coprocesseur Intel Xeon Phi

Le second type d’architecture machine à mémoire partagée concerne la configuration *many*-cœurs *Xeon Phi*. Ce coprocesseur offre 244 threads pour 61 cœurs CPU. Comme un thread est dédié à la gestion des interactions avec le nœud hôte (voir [JR13]), les expériences se bornent à 60 cœurs CPU, soit 240 threads. De plus, les résultats sont affichés en fonction du nombre de threads de telle sorte que, de 1 à 60, un thread est attribué à chaque cœur CPU, puis l’*hyperthreading* est activé, seulement par 2 jusqu’à 120 threads, et par 4 jusqu’à 240. Ce qui signifie par exemple pour 120 threads que chaque cœur CPU est partagé par 2 threads.

La Fig. 7.7 regroupe les différents *speedup* obtenus en faisant varier la granularité, sur les deux maillages, grossier en 7.7(a) et fin en 7.7(b). En gardant la granularité du code MPI, courbe rouge trait plein, le gain est nul, excepté sur 240 threads, ce que nous n’avons pas étudié en détails. L’utilisation de PaRSEC offre ici un gain intéressant, jusqu’à 1,6 de *speedup* pour 60 cœurs. La diminution de la granularité permet de meilleures performances, jusqu’à une certaine limite. En présence d’*hyperthreading*, c’est même la plus grande granularité (mais toujours plus fine que le code MPI) qui occasionne le plus fort gain. Une fois de plus, nous n’avons pas regardé en détails le comportement à 240 threads. L’*hyperthreading* est un problème à part entière et tous les codes ne se comportent pas de la même manière en l’utilisant. Ce sont donc les résultats jusqu’à 60 threads qui nous intéressent davantage. Nous avons néanmoins décidé de laisser l’ensemble des résultats, ce qui montre par ailleurs qu’avec l’*hyperthreading*, l’utilisation de PaRSEC reste bénéfique sur Intel Xeon Phi.

La Fig. 7.8 résume les résultats obtenus avec la meilleure granularité pour chaque expérience. Le *speedup* est présenté Fig. 7.8(a) et l’efficacité Fig. 7.8(b). Le code MPI a une efficacité décroissante au fur et à mesure que le nombre de cœurs CPU augmente, jusqu’à 60 threads. À l’inverse, le code utilisant PaRSEC produit une efficacité constante, très proche de 1, ce qui traduit une utilisation adéquate des ressources disponibles. L’ensemble des courbes décroche sévèrement après 60 threads, ce qui est normal car la formule appliquée ne tient pas compte de l’*hyperthreading*.

En effet, cette efficacité est mesurée par rapport au temps d’un thread disposant d’un cœur CPU pour lui seul. Or, l’*hyperthreading* signifie que plusieurs threads se partagent un cœur CPU. Il convient donc de mesurer l’efficacité relative à l’*hyperthreading*, c’est-à-

7.2. Résultats numériques

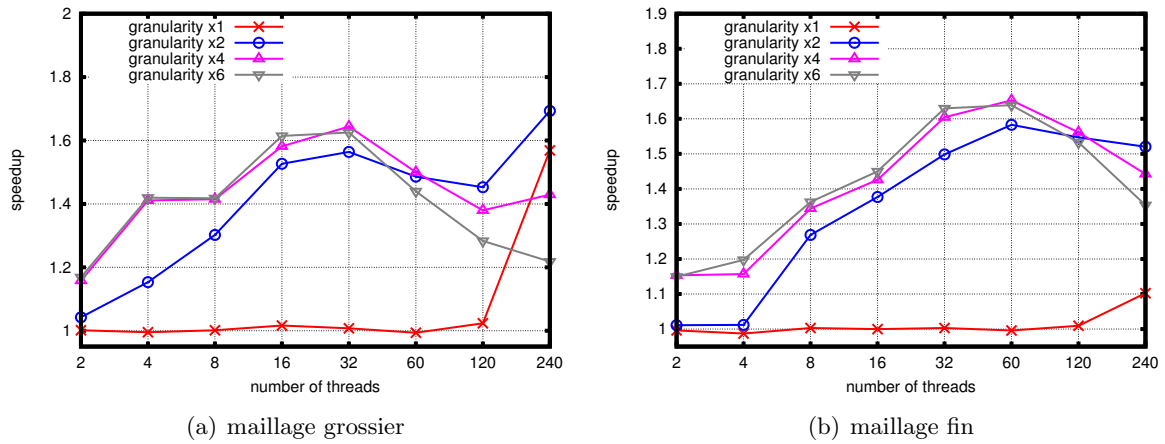


FIGURE 7.7 – *Speedup* sur *Xeon Phi*, pour différentes granularités

dire par rapport au temps que mettent 2 threads sur un cœur puis 4 threads sur un cœur. La tableau 7.4 présente cette efficacité relative. Le premier nombre correspond au maillage grossier et le second au maillage fin. Pour 120 threads, l'efficacité du code basé sur *PaRSEC* reste bien supérieure au code de référence *MPI* et toujours proche de 1. Pour 240 threads par contre, les efficacités sont dégradées pour les deux codes, avec toutefois un léger avantage à la configuration utilisant *PaRSEC*.

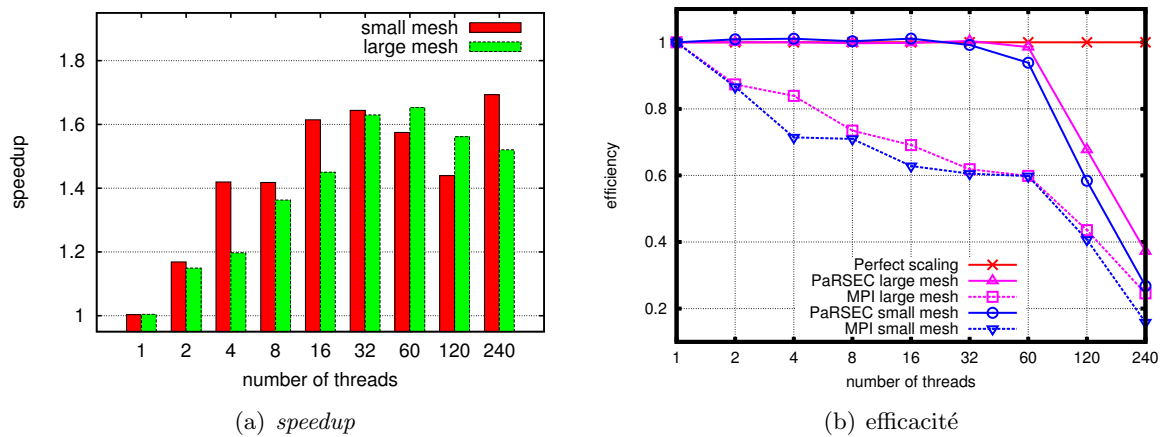


FIGURE 7.8 – Résultats *Xeon Phi*, meilleur *speedup* et efficacité

Nombre de threads	MPI		PaRSEC	
	maillage grossier	maillage fin	maillage grossier	maillage fin
120	0.66	0.71	0.80	0.91
240	0.53	0.80	0.62	0.84

TABLE 7.4 – Efficacité relative à l'*hyperthreading* pour la machine *Xeon Phi*

7.3 Prochaines étapes

Le projet est toujours en cours de développement, avec des objectifs à court terme et à moyen terme. Ce dernier point concerne les optimisations du JDF de la description du graphe de tâches, comme par exemple, l'ajout de priorités ou de poids sur les tâches, le changement d'heuristiques pour l'ordonnancement, ... Les autres points se focalisent essentiellement sur l'adressage de l'architecture machine à mémoire distribuée et l'utilisation détournée de l'*hyperthreading* des coprocesseurs Intel Xeon Phi.

Architectures à mémoire distribuée

Le choix d'adresser en premier lieu les architectures machines à mémoire partagée est clairement lié au souhait d'utiliser les coprocesseurs *many-cœurs* mais aussi parce que ces architectures sont plus simples à considérer. Cependant, les choix de développement tels que la description du JDF ou la construction des VP, ont été généralisés pour les deux architectures, mémoire partagée et distribuée. Ceci devrait limiter le temps nécessaire au portage du code pour les machines à mémoire distribuée.

De plus, la subdivision des sous-domaines qui sert à faire diminuer la granularité est prévue pour un nombre quelconque de sous-subdivisions. Ainsi, chaque sous-domaine de la première subdivision est associé à un nœud. Puis une deuxième subdivision répartit la charge de travail sur les ressources disponibles au sein de chaque nœud. Enfin, une troisième subdivision permet à chaque ressource d'avoir plusieurs sous-domaines à traiter, favorisant une gestion performante par un support d'exécution.

Hyperthreading des Intel Xeon Phi

L'*hyperthreading* est une capacité permettant dans certains cas un gain de performance. Comme le nombre de ressources physiques reste le même, tous les codes n'en profitent pas. Pour cette raison, PaRSEC envisage d'utiliser l'*hyperthreading* pour améliorer les transferts de données locaux et distants plutôt que pour ajouter des ressources virtuelles pour du calcul. Ce qui peut occasionner un effet inverse en augmentant le temps d'exécution.

Chaque cœur CPU dispose d'un cache L2 propre mais il n'y a pas de cache L3 partagé comme sur les cœurs CPU classiques. De plus, les mouvements de données entre ces caches se font à travers un anneau bidirectionnel très rapide, ce qui bouleverse les techniques d'optimisations classiques. Pour cela, il n'est pas toujours efficace d'utiliser des *virtual processes* lors de l'adressage d'un coprocesseur Intel Xeon Phi, comme dans notre cas par exemple.

Conclusion à la partie II

L'évolution des supercalculateurs a suivi la loi de Moore au prix d'une complexification des architectures : multi-processeurs et *hyperthreading* au début des années 2000, accélérateurs GPU et coprocesseurs à la fin des années 2000. Quant à l'avenir, les processeurs s'orientent déjà vers une architecture *many*-cœurs pour relever le défi de l'*exascale* en 2020. Cela conduit à des machines composées de différentes unités de calcul, possédant chacune leurs propres niveaux de mémoires, formant ce que l'on appelle l'hétérogénéité. La problématique vient du fait qu'il est difficile pour un programmeur d'optimiser un code informatique exploitant efficacement ces machines. En effet, il faut répartir intelligemment le travail sur les ressources disponibles, qui n'ont pas toutes les mêmes capacités, tout en se chargeant des transferts mémoires distants et des optimisations de mémoire cache. Cela demande beaucoup de temps et s'avère le plus souvent très dépendant de la machine ciblée, rendant le code non-portable.

Les bibliothèques d'algèbre linéaire sont très utilisées en simulation numérique et sur tous les types de machines. Pour conserver un code portable, tout en exploitant efficacement les machines hétérogènes, certaines d'entre-elles ont opté pour le paradigme de parallélisme à base de tâches. Dans ce modèle, le flot d'exécution est décrit dans un arbre de tâches dont les nœuds représentent les fonctions et les arêtes sont les dépendances de données entre ces fonctions. Il n'y a plus de référence directe à l'architecture sous-jacente, comme dans un pseudo-code mathématique.

L'exécution d'un programme à base de tâches a vocation à être supportée par un moteur d'exécution. Son rôle est d'orchestrer la réalisation des tâches en respectant leurs dépendances. Pour être efficace, un support d'exécution se sert de politiques d'ordonnancement statiques ou dynamiques. Certains se chargent également des transferts mémoires distants et peuvent aussi adresser des accélérateurs. C'est le cas de PaRSEC, développé à l'*Innovative Computing Laboratory, University of Tennessee, USA*) par l'équipe de George BOSILCA. C'est un exemple de support d'exécution polyvalent, stable et mature.

Le code de propagation des ondes DIVA de Total repose sur une discrétisation par la méthode d'éléments finis DG associée à un schéma en temps Leap-Frog. Nous travaillons avec la formulation du premier ordre qui fait apparaître deux variables, le vecteur vitesse et le tenseur des contraintes. Le fait que des matrices de masses DG sont diagonales par blocs permet d'écrire le système matriciel sous une forme quasi-explicite (une inversion de matrice locale restant à faire à l'initialisation).

L'algorithme du pseudo-code parallèle est basé sur une décomposition de domaines. Dans l'implémentation initiale, reposant sur le paradigme par passage de message de la bibliothèque MPI, le nombre de sous-domaines est alors égal au nombre de processus MPI, c'est-à-dire au nombre d'unités de calculs CPU. À chaque itération, comme la mise à jour d'une variable

dépend de l'autre variable, les calculs sont effectués dans deux boucles différentes sur les sous-domaines, une par variable, chacune après une étape de communication des données à l'interface des sous-domaines.

Un avantage de la méthode DG est de pouvoir représenter des zones de calcul avec plus ou moins de degrés de liberté. Cela conduit à des maillages hétérogènes où la taille des variables est différente selon les mailles. Il est possible de calculer le nombre exact d'opérations dans chaque sous-domaine mais cela ne donne pas le temps exact des calculs qui dépend des capacités des ressources informatiques. Nous avons illustré ce problème d'équilibrage de charge pour le code DIVA ainsi que la décroissance de son efficacité parallèle.

La solution que nous avons proposée est de changer le paradigme de parallélisme en choisissant la programmation à base de tâches. La faisabilité d'un tel portage, pour un code à l'échelle industrielle, restait à démontrer. Dans le cadre du partenariat DIP, nous avons profité de la venue de George BOSILCA pour mener ce projet en collaboration et nous avons donc choisi le support d'exécution PaRSEC. Nous avons ainsi contribué également à son développement en améliorant la prise en charge des codes en Fortran.

Nous avons commencé par écrire le graphe de tâches associé au code DIVA. Chaque étape de calculs amène deux tâches de mise à jour des variables. Il n'y a plus de notion de communication mais il faut néanmoins préparer les données à transmettre avec les autres sous-domaines et répartir les données qui proviennent de ces derniers. Cela crée un nombre de tâches par itération égal à quatre fois le nombre de sous-domaines voisins par sous-domaine.

Nous avons ensuite décrit ces tâches suivant un modèle paramétré et plus spécifiquement en JDF (*Job Data Flow*), l'extension propre de PaRSEC pour la description des tâches. Il faut et il suffit de décrire pour chaque donnée de chaque tâche, la donnée et la tâche en provenance et la donnée et la tâche en destination. Le mode de traitement de ces données est aussi à préciser : lecture, écriture ou lecture-écriture. Cela permet d'optimiser l'ordonnancement qui respecte la cohérence des données.

Enfin, nous avons diminué la granularité des tâches en les rendant plus petites. Pour cela nous avons modifié la décomposition de domaines pour ajouter une étape de subdivision supplémentaire. Ceci nous a permis de garder une certaine localité des sous-domaines. En effet, il y a maintenant plus de sous-domaines que d'unités de calcul et chaque unité à une préférence pour un groupe de sous-domaine contigus. La granularité fine exhibe plus de parallélisme et, associée au vol de tâches, permet au support d'exécution d'optimiser l'ordre de réalisation des tâches. Nous avons illustré ces points clés sur un processeur, sur lequel nous avons déjà obtenu un gain par rapport au code initial.

Nous avons démontré la portabilité des performances du code DIVA en tâches sur des architectures particulières, comme des machines ccNUMA (*cache coherent Non-Uniform Memory Access*) ou des coprocesseurs Intel Xeon Phi. L'étude expérimentale a montré que les meilleurs résultats sont obtenus avec une granularité beaucoup fine que celle du code d'origine.

- Pour la machine ccNUMA, la difficulté vient du facteur NUMA qui correspond au coût d'accès variable à la mémoire locale disposée en bancs. Pour cela nous avons utilisé les *virtual processes* de PaRSEC qui permettent d'agréger virtuellement des unités de calculs pour y définir une priorité de traitement. Nous avons ainsi obtenu un *speedup* par rapport au code initial et une efficacité parallèle qui reste proche de 1 et qui décroît légèrement pour 64 cœurs.

- Pour le coprocesseur Intel Xeon Phi, les performances théoriques maximales reposent sur l'utilisation poussée de l'*hyperthreading*. Nous ne savons pas si le code DIVA peut tirer profit de ce mode d'exécution. Sans l'*hyperthreading*, c'est-à-dire jusqu'à 60 threads, nous avons obtenu un *speedup* intéressant par rapport au code initial et une efficacité parallèle qui reste très proche de 1. Avec l'*hyperthreading*, nous avons également obtenu un *speedup* mais moins important et une efficacité parallèle (relative) qui décroît.

Ces travaux ont été présentés dans plusieurs conférences internationales : YIC ECCOMAS2014, IEEE HPCC2014 et EAGE HPC2014 notamment. Un résumé a été accepté pour publication : [BBAC14]. D'un point de vue industriel, ce travail a permis de démontrer l'efficacité de la programmation à base de tâches aussi bien en terme de productivité que de performance et Total envisage de porter d'autres codes sur ce paradigme.

Pour aller plus loin, il reste à considérer en premier lieu des architectures à mémoire distribuée. Pour cela, quelques modifications restent à apporter mais le travail déjà effectué sur la subdivision du maillage ou l'utilisation des *virtual processes* a été généralisé pour prendre en compte un niveau supplémentaire de hiérarchie mémoire. Nous aimerions compléter l'illustration de la portabilité du code DIVA en tâches et étudier son passage à l'échelle sur beaucoup plus de cœurs.

Il nous semble par ailleurs intéressant de faire une étude de l'impact général des conditions aux limites sur un code parallèle dans un contexte HPC. La comparaison peut alors être faite sur les différentes conditions aux limites classiquement utilisées mais aussi sur leurs comportements selon l'implémentation choisie, en MPI ou en tâches. Nous envisageons également de mieux revoir la prise en compte de l'*hyperthreading* afin d'augmenter l'efficacité du code sur les architectures *many*-cœurs.

Annexe A

Annexes à la partie I

A.1 Rotation des matrices d'élasticité anisotrope

L'anisotropie TTI est une inclinaison par les angles (1.23) de l'anisotropie VTI, voir Fig. 1.4. La matrice d'élasticité TTI exprimée dans le repère d'orthotropie $(\tilde{e}_1, \tilde{e}_2, \tilde{e}_3)$ est simplement VTI. De là, il est possible de formuler une première rotation suivant un des axes du repère d'orthotropie, puis une seconde rotation pour se retrouver dans le repère cartésien (e_1, e_2, e_3) .

L'axe \tilde{e}_2 fait partie du plan (e_1, e_2) . La rotation selon l'angle dip $\theta = \widehat{(e_3, \tilde{e}_3)}$ dans le plan $(\tilde{e}_1, \tilde{e}_3)$ s'écrit par la matrice de passage du repère d'orthotropie $(\tilde{e}_1, \tilde{e}_2, \tilde{e}_3)$ au repère $(P_{12}\tilde{e}_3, \tilde{e}_2, e_3)$:

$$M_\theta = \begin{bmatrix} \cos^2 \theta & 0 & \sin^2 \theta & -2 \cos \theta \sin \theta & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \sin^2 \theta & 0 & \cos^2 \theta & 2 \cos \theta \sin \theta & 0 & 0 \\ \cos \theta \sin \theta & 0 & -\cos \theta \sin \theta & \cos^2 \theta - \sin^2 \theta & 0 & 0 \\ 0 & 0 & 0 & 0 & -\sin \theta & \cos \theta \\ 0 & 0 & 0 & 0 & \cos \theta & \sin \theta \end{bmatrix} \quad (\text{A.1})$$

où P_{12} est la projection dans le plan d'axes (e_1, e_2) .

Puis, la rotation selon l'angle azimut $\phi = \widehat{(e_1, P_{12}\tilde{e}_3)}$ dans le plan (e_1, e_2) s'écrit par la matrice de passage du repère $(P_{12}\tilde{e}_3, \tilde{e}_2, e_3)$ au repère cartésien (e_1, e_2, e_3) :

$$M_\phi = \begin{bmatrix} \cos^2 \phi & \sin^2 \phi & 0 & 0 & 0 & -2 \cos \phi \sin \phi \\ \sin^2 \phi & \cos^2 \phi & 0 & 0 & 0 & 2 \cos \phi \sin \phi \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & -\sin \phi & \cos \phi & 0 \\ \cos \phi \sin \phi & -\cos \phi \sin \phi & 0 & 0 & 0 & \cos^2 \phi - \sin^2 \phi \end{bmatrix} \quad (\text{A.2})$$

La matrice d'élasticité TTI est le résultat du produit de la matrice d'élasticité VTI par les matrices de passage M_θ et M_ϕ , selon la formule :

$$M_\phi M_\theta C_{VTI} M_\theta^T M_\phi^T$$

A.2 Code Maple™ pour les valeurs propres et vecteurs propres VTI 2D

```
#####
# Détermination des valeurs propres et des vecteurs propres #
# pour les ondes élastiques en VTI 2D. #
#####

#####
# valeurs propres #
#####

# structure de la matrice de l'EDO en W
restart;
with(LinearAlgebra);
M:= Matrix([[0,0,-1,0], [0,0,0,-1], [a,0,0,c], [0,b,d,0]]);
eVal:=Eigenvalues(M):

# valeurs propres calculées
lp:=eVal(1);
# = (1/2)*sqrt(-2*b+2*d*c-2*a+2*sqrt(b^2-2*b*d*c-2*a*b+d^2*c^2-2*d*a*c+a^2))
ls:=eVal(3);
# = (1/2)*sqrt(-2*b+2*d*c-2*a-2*sqrt(b^2-2*b*d*c-2*a*b+d^2*c^2-2*d*a*c+a^2))

# sachant
a:=(rho*omega^2-xi^2*C66)/C11;
b:=(rho*omega^2-xi^2*C33)/C66;
c:=-I*xi*(C13+C66)/C11;
d:=-I*xi*(C13+C66)/C66;

# Notre proposition de valeurs propres
test_lp:=sqrt(alpha*xi^2+beta*rho*omega^2\
+sqrt(gam*xi^4+eta*rho*omega^2*xi^2+nu*rho^2*omega^4));
test_ls:=sqrt(alpha*xi^2+beta*rho*omega^2\
-sqrt(gam*xi^4+eta*rho*omega^2*xi^2+nu*rho^2*omega^4));
# sachant
alpha:=-(C13^2-C11*C33+2*C13*C66)/(2*C11*C66);
beta:=-(C11+C66)/(2*C11*C66);
gam:=alpha^2-C33/C11;
eta:=2*alpha*beta+(C33+C66)/(C11*C66);
nu:=((C11-C66)/(2*C11*C66))^2;

## Vérification par identification
# première partie de la racine carrée
simplify((alpha*xi^2+beta*rho*omega^2)-(-2*b+2*d*c-2*a)/4);
# = 0
```

A.2. Code Maple™ pour les valeurs propres et vecteurs propres VTI 2D

```
#deuxième partie de la racine carrée test_lp-lp
simplify((gam*xi^4+eta*rho*omega^2*xi^2+nu*rho^2*omega^4)\
-(b^2-2*b*d*c-2*a*b+d^2*c^2-2*d*a*c+a^2)/4);
# = 0

#####
# vecteurs propres #
#####

# structure de la matrice de l'EDO en W
restart;
with(LinearAlgebra);
M:= Matrix([[0,0,-1,0], [0,0,0,-1], [a,0,0,c], [0,b,d,0]]);
l,L:=Eigenvectors(M):
lp:=l(1):
ls:=l(3):

# Soient
Xp:=-I*xi*(-c*b/(lp^2*(b+ls^2)));
Xs:=-c*b/((I*xi)*(b+lp^2));
# ou encore
# Xs:=-c*b/((I*xi)*(a-c*d+ls^2));

# Notre proposition de matrice de vecteurs propres
L2:=Matrix([[-Xp/(I*xi),-Xp/(I*xi),Xs*I*xi/ls^2,Xs*I*xi/ls^2],[-1/lp,1/lp,-1/ls,1/ls],\
[Xp*lp/(I*xi),-Xp*lp/(I*xi),-Xs*I*xi/ls,Xs*I*xi/ls],[1,1,1,1]]);

## Vérification
simplify(L-L2);
# = 0

####
```


A.3 Code Maple™ pour la simplification à l'ordre 0 d'un système

```
#####
# Simplification à l'ordre 0 autour de  $x^2 = \xi^2 / \omega^2$  #
# de  $s_{xx} = C_{11}*(A*X_p*lp^2 - B*X_s*I*\xi*ls) - I*\xi*C_{13}*v_z$  #
# et  $s_{xz} = C_{66}*(-A*x*I*\xi*lp + B*x*ls^2) - I*\xi*C_{66}*v_x$  #
#####

### constantes du système
a := (rho*omega^2-xi^2*C66)/C11;
b := (rho*omega^2-xi^2*C33)/C66;
c := -I*xi*(C13+C66)/C11;
d := -I*xi*(C13+C66)/C66;

### valeurs propres
# qui peuvent se réécrire directement
lp := -I*omega*sqrt(-alpha*x^2-beta*rho-sqrt(gam*x^4+eta*rho*x^2+nu*rho^2));
ls := -I*omega*sqrt(-alpha*x^2-beta*rho+sqrt(gam*x^4+eta*rho*x^2+nu*rho^2));

### facteurs
Xp := -I*xi*(-c*b/(lp^2*(b+ls^2)));
Xs := -c*b/((I*xi)*(b+lp^2));
A := (-ls*vz-I*xi*Xs*vz)/(Xs*xi^2-Xp*ls*lp);
B := (-I*xi*vz+Xp*lp*vz)/(Xs*xi^2-Xp*ls*lp);
# on isole les parties portant sur v_x et sur v_z
Ax:=eval(A,[vx=1,vz=0]);
Bx:=eval(B,[vx=1,vz=0]);
Az:=eval(A,[vx=0,vz=1]);
Bz:=eval(B,[vx=0,vz=1]);

### systeme à simplifier, décomposé en x et z
sxx_x:=C11*(-Ax*Xp*lp^2-Bx*Xs*I*xi*ls):
sxx_z:=C11*(-Az*Xp*lp^2-Bz*Xs*I*xi*ls)-I*xi*C13:
sxz_x:=C66*(-Ax*I*xi*lp+Bx*ls^2):
sxz_z:=C66*(-Az*I*xi*lp+Bz*ls^2):

### marche à suivre pour reproduire les résultats
# pour plus de maniabilité dans Maple, l'étude est faite terme par terme
# puis ces termes sont copier-coller en remplaçant  $\xi^2/\omega^2$  par  $x^2$ 
# enfin, la fonction "taylor(...,x,1)" est appelée sur chaque composante
# il suffit ensuite de faire la somme des deux composantes puis d'évaluer :
# "eval(...,[beta=-(C11+C66)/(2*C11*C66),nu=((C11-C66)/(2*C11*C66))^2])"
# sans oublier les hypothèse physiques "assuming C11>0, C66>0, C11>C66"

### sxx_x
```

A.3. Code Maple™ pour la simplification à l'ordre 0 d'un système

```
# garder -I*omega en facteur
sxx_x_1:=C11*(-Ax*Xp*lp^2);
sxx_x_2:=C11*(-Bx*Xs*I*xi*ls);
# = sqrt(rho)*I*omega*sqrt(C11)

### sxx_z
# garder I*xi en facteur
sxx_z_1:=C11*(-Az*Xp*lp^2);
sxx_z_2:=C11*(-Bz*Xs*I*xi*ls);
# -I*xi*C13
# = I*xi*(C13+C66)*(C11-sqrt(C66)*sqrt(C11))/(C11-C66)-I*xi*C13

### sxz_x
# garder -I*omega^2/xi en facteur
sxz_x_1:=C66*(-Ax*I*xi*lp);
sxz_x_2:=C66*(Bx*ls^2);
# -I*xi*C66
# = 0-I*xi*C66

### sxz_z
# garder -I*omega en facteur
sxz_z_1:=C66*(-Az*I*xi*lp);
sxz_z_2:=C66*(Bz*ls^2);
# = sqrt(rho)*I*omega*sqrt(C66)

####
```

A.4 Code Maple™ pour les coefficients de réflexion VTI 2D

```
#####
# Coefficient de réflexion pour la CLA VTI 2D #
#####

#####
# ondes P #
#####

restart;
### décomposition du champ total
VecI := Vector([X*kx,Z*kz]):
Vec_i := Vector([VecI(1), VecI(2)]):
Vec_i := subs([kx = kx_i, kz = kz_i], Vec_i):
Vec_j := Vector([-VecI(2), VecI(1)]):
Vec_j := subs([kx = kx_j, kz = kz_j], Vec_j):

### équation sur le bord VTI
U := VecI+Aii*Vec_i+Aij*Vec_j:
dxU := kx*VecI+kx_i*Aii*Vec_i+kx_j*Aij*Vec_j:
dzU := kz*VecI+kz_i*Aii*Vec_i+kz_j*Aij*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2),C66*(dxU(2)+dzU(1))]):
B := Vector([rho*kappa*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii, Aij}):
assign(%):

### force CLA isotrope
U := VecI+Aii_iso*Vec_i+Aij_iso*Vec_j:
dxU := kx*VecI+kx_i*Aii_iso*Vec_i+kx_j*Aij_iso*Vec_j:
dzU := kz*VecI+kz_i*Aii_iso*Vec_i+kz_j*Aij_iso*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2),C66*(dxU(2)+dzU(1))]):
B := Vector([rho*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_iso, Aij_iso}):
assign(%):

### paramètres
X:=sqrt(kappa^2*Vp^2-Vs^2):
Z:=sqrt(Vp^2-Vs^2):
C11:=rho*kappa^2*Vp^2:
C66:=rho*Vs^2:
C13:=rho*X*Z-rho*Vs^2:

### rapport k/w
kx:=cos(theta)/(kappa*Vp):
kx_i:=-cos(theta)/(kappa*Vp):
kx_j:=-sqrt(1/Vs^2-(sin(theta)^2)/(Vp^2)):
```

```

kz:=sin(theta)/Vp:
kz_i:=kz:
kz_j:=kz:
App := eval(Aii): Aps := eval(Aij):
App_iso := eval(Aii_iso): Aps_iso := eval(Aij_iso):
'assuming'([simplify(taylor(App, theta = 0, 3))], [Vs::positive]);
'assuming'([simplify(taylor(App_iso, theta = 0, 3))], [Vs::positive]);
'assuming'([simplify(taylor(Aps, theta = 0, 3))], [Vs::positive]);

### figure
App_VTI:=eval(abs(App), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
App_VTI_iso:=eval(abs(App_iso), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
App_ref_iso:=eval(abs(App), [kappa = 1, Vp = 3000, Vs = 1750]):
plot([App_VTI, App_VTI_iso, App_ref_iso], theta = 0 .. (1/2)*Pi, linestyle = [1, 3, 4]);
Aps_VTI:=eval(abs(Aps), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Aps_VTI_iso:=eval(abs(Aps_iso), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Aps_ref_iso:=eval(abs(Aps), [kappa = 1, Vp = 3000, Vs = 1750]):
plot([Aps_VTI, Aps_VTI_iso, Aps_ref_iso], theta = 0 .. (1/2)*Pi, linestyle = [1, 3, 4]);

#####
# ondes S #
#####

restart;
### décomposition du champ total
VecI := Vector([-Z*kz,X*kx]):
Vec_i := Vector([VecI(1), VecI(2)]):
Vec_i := subs([kx = kx_i, kz = kz_i], Vec_i):
Vec_j := Vector([-VecI(2), VecI(1)]):
Vec_j := subs([kx = kx_j, kz = kz_j], Vec_j):

### équation sur le bord VTI
U := VecI+Aii*Vec_i+Aij*Vec_j:
dxU := kx*VecI+kx_i*Aii*Vec_i+kx_j*Aij*Vec_j:
dzU := kz*VecI+kz_i*Aii*Vec_i+kz_j*Aij*Vec_j:
B := Vector([rho*kappa*Vp*U(1),rho*Vs*U(2)]):
E := Vector([C11*dxU(1)+C13*dzU(2),C66*(dxU(2)+dzU(1))]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii, Aij}):
assign(%):

### force CLA isotrope
U := VecI+Aii_iso*Vec_i+Aij_iso*Vec_j:
dxU := kx*VecI+kx_i*Aii_iso*Vec_i+kx_j*Aij_iso*Vec_j:
dzU := kz*VecI+kz_i*Aii_iso*Vec_i+kz_j*Aij_iso*Vec_j:
B := Vector([rho*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_iso, Aij_iso}):
assign(%):

```

```

### paramètres
X:=sqrt(kappa^2*Vp^2-Vs^2):
Z:=sqrt(Vp^2-Vs^2):
C11:=rho*kappa^2*Vp^2:
C66:=rho*Vs^2:
C13:=rho*X*Z-rho*Vs^2:

### rapports k/w
kx:=cos(theta)/Vs:
kx_i:=-cos(theta)/Vs:
kx_j:=-sqrt(1/(kappa^2*Vp^2)-(sin(theta)^2)/(kappa^2*Vs^2)):
kz:=sin(theta)/Vs:
kz_i:=kz:
kz_j:=kz:
Ass := eval(Aii): Asp := eval(Aij):
Ass_iso := eval(Aii_iso): Asp_iso := eval(Aij_iso):
'assuming'([simplify(taylor(Ass, theta = 0, 3))], [Vp::positive,kappa::positive]);
'assuming'([simplify(taylor(Asp, theta = 0, 3))], [Vp::positive,kappa::positive]);

### figure
Ass_VTI:=eval(abs(Ass), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Ass_VTI_iso:=eval(abs(Ass_iso), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Ass_ref_iso:=eval(abs(Ass), [kappa = 1, Vp = 3000, Vs = 1750]):
plot([Ass_VTI, Ass_VTI_iso, Ass_ref_iso], theta = 0 .. (1/2)*Pi, linestyle = [1, 3, 4]);
Asp_VTI:=eval(abs(Asp), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Asp_VTI_iso:=eval(abs(Asp_iso), [kappa = sqrt(1.5), Vp = 3000, Vs = 1750]):
Asp_ref_iso:=eval(abs(Asp), [kappa = 1, Vp = 3000, Vs = 1750]):
plot([Asp_VTI, Asp_VTI_iso, Asp_ref_iso], theta = 0 .. (1/2)*Pi, linestyle = [1, 3, 4]);

####

```

A.5 Code Maple™ pour les courbes de lenteur

```
#####
# Courbes de lenteur en TTI 2D #
#####

restart:
with(LinearAlgebra):

### paramètres TTI
rho:=1:
Vp:=3000:
Vs:=1800:
epsilon:=0.20:
delta:=0.15:
theta_TTI:=30*Pi/180:

Cvti:=Matrix(3,3):
Cvti[1,1]:=rho*Vp^2*(1+2*epsilon):
Cvti[1,2]:=sqrt(rho^2*(Vp^2-Vs^2)^2+2*delta*Vp^2*rho^2*(Vp^2-Vs^2)) - rho*Vs^2:
Cvti[2,2]:=rho*Vp^2:
Cvti[3,3]:=rho*Vs^2:

### TTI par rotation du VTI
C:=Matrix(3,3):
R := Matrix([[cos(theta_TTI),sin(theta_TTI)],[-sin(theta_TTI),cos(theta_TTI)]]):
V := Matrix([[1,3],[3,2]]):

Cvti[1,3]:=0:
Cvti[2,3]:=0:
Cvti[2,1]:=Cvti[1,2]:
Cvti[3,1]:=Cvti[1,3]:
Cvti[3,2]:=Cvti[2,3]:
for i from 1 to 2 do
  for j from 1 to 2 do
    for k from 1 to 2 do
      for l from 1 to 2 do
        temp:=0:
        for p from 1 to 2 do
          for q from 1 to 2 do
            for r from 1 to 2 do
              for s from 1 to 2 do
                temp:=temp+R[p,i]*R[q,j]*R[r,k]*R[s,l]*Cvti[V[p,q],V[r,s]]:
              od;
            od;
          od;
        od;
      od;
    od;
  od;
end do;
```

```
        C[V[i,j],V[k,l]]:=temp:
    od;
od;
od;
od;

### équation de dispersion
A:=Matrix(2,2):
A[1,1]:=C[1,1]*x^2+2*C[1,3]*x*y+C[3,3]*y^2:
A[1,2]:=C[1,3]*x^2+(C[3,3]+C[1,2])*x*y+C[2,3]*y^2:
A[2,1]:=A[1,2]:
A[2,2]:=C[3,3]*x^2+2*C[2,3]*x*y+C[2,2]*y^2:
eVal:=Eigenvalues(evalf(A)):

### courbes de lenteur
slow_p:=subs(x=cos(theta),y=sin(theta),sqrt(rho/eVal[1])):
slow_s:=subs(x=cos(theta),y=sin(theta),sqrt(rho/eVal[2])):
plot([slow_p,slow_s],theta=0..2*Pi,coords=polar,scaling=CONSTRAINED,linestyle=[1,3]);
```

A.6 Code Maple™ pour les coefficients de réflexion TTI 2D

```
#####
# Coefficient de réflexion pour la CLA TTI 2D #
#####

restart;

rho:=1: Vp:=3000: Vs:=1750: kappa:=sqrt(1.5): alpha:=30*Pi/180:

### paramètres
X:=sqrt(kappa^2*Vp^2-Vs^2):
Z:=sqrt(Vp^2-Vs^2):
Cvti[1,1]:=Vp^2*kappa^2;
Cvti[1,2]:=X*Z-Vs^2;
Cvti[2,2]:=Vp^2;
Cvti[3,3]:=Vs^2;
C:=Matrix(3,3):
V := Matrix([[1,3],[3,2]]):
R := Matrix([[cos(alpha),-sin(alpha)],[sin(alpha),cos(alpha)]]):
Cvti[1,3]:=0
Cvti[2,3]:=0:
Cvti[2,1]:=Cvti[1,2]:
Cvti[3,1]:=Cvti[1,3]:
Cvti[3,2]:=Cvti[2,3]:
for i from 1 to 2 do
  for j from 1 to 2 do
    for k from 1 to 2 do
      for l from 1 to 2 do
        temp:=0:
        for p from 1 to 2 do
          for q from 1 to 2 do
            for r from 1 to 2 do
              for s from 1 to 2 do
                temp:=temp+R[p,i]*R[q,j]*R[r,k]*R[s,l]*Cvti[V[p,q],V[r,s]]:
              od;
            od;
          od;
        od;
        C[V[i,j],V[k,l]]:=temp:
      od;
    od;
  od;
od;
C11:=C(1,1):
C13:=C(1,2):
C66:=C(3,3):
```


Annexe A. Annexes à la partie I

```
C16:=C(1,3):
C36:=C(2,3):
xi_1:=rho*Vp^2*(kappa^2*cos(alpha)^2+sin(alpha)^2):
xi_2:=-rho*Vp^2*(kappa^2-1)*2*cos(alpha)*sin(alpha):
xi_3:=rho*Vp^2*(kappa^2*sin(alpha)^2+cos(alpha)^2):

#####
# ondes P # (si vous venez de calculer l'onde S, faites le "restart")
#####

### décomposition du champ total
VecI := Vector([(X*cos(alpha)^2+Z*sin(alpha)^2)*kx+(X-Z)*cos(alpha)*sin(alpha)*kz, \
(Z*cos(alpha)^2+X*sin(alpha)^2)*kz+(X-Z)*cos(alpha)*sin(alpha)*kx]):
Vec_i := Vector([VecI(1), VecI(2)]):
Vec_i := subs([kx = kx_i, kz = kz_i], Vec_i):
Vec_j := Vector([-VecI(2), VecI(1)]):
Vec_j := subs([kx = kx_j, kz = kz_j], Vec_j):

### équation sur le bord TTI
U := VecI+Aii*Vec_i+Aij*Vec_j:
dxU := kx*VecI+kx_i*Aii*Vec_i+kx_j*Aij*Vec_j:
dzU := kz*VecI+kz_i*Aii*Vec_i+kz_j*Aij*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
B1 := kappa*cos(alpha)^2+sin(alpha)^2:
B2 := -(kappa-1)*cos(alpha)*sin(alpha):
B3:= sqrt(kappa^2*cos(alpha)^2+sin(alpha)^2):
B := Vector([B1^2/B3*Vp*U(1)+B1*B2/B3*Vp*U(2), B1*B2/B3*Vp*U(1)+B2^2/B3*Vp*U(2)+Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii, Aij}):
assign(%):

### force CLA VTI
U := VecI+Aii_VTI*Vec_i+Aij_VTI*Vec_j:
dxU := kx*VecI+kx_i*Aii_VTI*Vec_i+kx_j*Aij_VTI*Vec_j:
dzU := kz*VecI+kz_i*Aii_VTI*Vec_i+kz_j*Aij_VTI*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
B := Vector([rho*kappa*Vp*U(1), rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_VTI, Aij_VTI}):
assign(%):

### force CLA iso
U := VecI+Aii_iso*Vec_i+Aij_iso*Vec_j:
dxU := kx*VecI+kx_i*Aii_iso*Vec_i+kx_j*Aij_iso*Vec_j:
dzU := kz*VecI+kz_i*Aii_iso*Vec_i+kz_j*Aij_iso*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
```

A.6. Code Maple™ pour les coefficients de réflexion TTI 2D

```

B := Vector([rho*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_iso, Aij_iso}):
assign(%):

### rapport k/w
# kz devient ici le paramètre d'observation
kz_i:=kz: kz_j:=kz:
kx1, kx2 := solve(kx^2*xi_1+kx*kz*xi_2+kz^2*xi_3 = 1, kx):
kx:=kx1:
kx_i:=kx2:
kx_j:=-sqrt(1/(Vs^2)-kz^2):

### figure
App := eval(Aii): Aps := eval(Aij):
App_VTI := eval(Aii_VTI): Aps_VTI := eval(Aij_VTI):
App_iso := eval(Aii_iso): Aps_iso := eval(Aij_iso):
kz1:=-2*sqrt(xi_1)/sqrt(4*xi_1*xi_3-xi_2^2): kz2:=2*sqrt(xi_1)/sqrt(4*xi_1*xi_3-xi_2^2):
plot([abs(App), abs(App_VTI), abs(App_iso)], kz=kz1-1e-6..kz2+1e-6, linestyle=[1, 3, 4]);
plot([abs(Aps), abs(Aps_VTI), abs(Aps_iso)], kz=kz1-1e-6..kz2+1e-6, linestyle=[1, 3, 4]);

#####
# ondes S # (si vous venez de calculer l'onde P, faites le "restart")
#####

### décomposition du champ total
VecI := Vector([- (Z*cos(alpha)^2+X*sin(alpha)^2)*kz-(X-Z)*cos(alpha)*sin(alpha)*kx, \
(X*cos(alpha)^2+Z*sin(alpha)^2)*kx+(X-Z)*cos(alpha)*sin(alpha)*kz]):
Vec_i := Vector([VecI(1), VecI(2)]):
Vec_i := subs([kx = kx_i, kz = kz_i], Vec_i):
Vec_j := Vector([-VecI(2), VecI(1)]):
Vec_j := subs([kx = kx_j, kz = kz_j], Vec_j):

### équation sur le bord TTI
U := VecI+Aii*Vec_i+Aij*Vec_j:
dxU := kx*VecI+kx_i*Aii*Vec_i+kx_j*Aij*Vec_j:
dzU := kz*VecI+kz_i*Aii*Vec_i+kz_j*Aij*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
B1 := kappa*cos(alpha)^2+sin(alpha)^2:
B2 := -(kappa-1)*cos(alpha)*sin(alpha):
B3:= sqrt(kappa^2*cos(alpha)^2+sin(alpha)^2):
B := Vector([B1^2/B3*Vp*U(1)+B1*B2/B3*Vp*U(2),B1*B2/B3*Vp*U(1)+B2^2/B3*Vp*U(2)+Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii, Aij}):
assign(%):

### force CLA VTI
U := VecI+Aii_VTI*Vec_i+Aij_VTI*Vec_j:

```

Annexe A. Annexes à la partie I

```
dxU := kx*VecI+kx_i*Aii_VTI*Vec_i+kx_j*Aij_VTI*Vec_j:
dzU := kz*VecI+kz_i*Aii_VTI*Vec_i+kz_j*Aij_VTI*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
            C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
B := Vector([rho*kappa*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_VTI, Aij_VTI}):
assign(%):

### force CLA iso
U := VecI+Aii_iso*Vec_i+Aij_iso*Vec_j:
dxU := kx*VecI+kx_i*Aii_iso*Vec_i+kx_j*Aij_iso*Vec_j:
dzU := kz*VecI+kz_i*Aii_iso*Vec_i+kz_j*Aij_iso*Vec_j:
E := Vector([C11*dxU(1)+C13*dzU(2)+C16*(dxU(2)+dzU(1)), \
            C16*dxU(1)+C36*dzU(2)+C66*(dxU(2)+dzU(1))]):
B := Vector([rho*Vp*U(1),rho*Vs*U(2)]):
solve({E(1) = B(1), E(2) = B(2)}, {Aii_iso, Aij_iso}):
assign(%):

### rapport k/w
kz:=sin(theta)/Vs:
kz_i:=kz:
kz_j:=kz:
kx1, kx2 := solve(kx^2*xi_1+kx*kz*xi_2+kz^2*xi_3 = 1, kx):
kx:=cos(theta)/Vs:
kx_i:=-cos(theta)/Vs:
kx_j:=kx2:

### figure
Ass := eval(Aii): Asp := eval(Aij):
Ass_VTI := eval(Aii_VTI): Asp_VTI := eval(Aij_VTI):
Ass_iso := eval(Aii_iso): Asp_iso := eval(Aij_iso):
plot([abs(Ass), abs(Ass_VTI), abs(Ass_iso)], theta=-Pi/2..Pi/2, linestyle = [1, 3, 4]);
plot([abs(Asp), abs(Asp_VTI), abs(Asp_iso)], theta=-Pi/2..Pi/2, linestyle = [1, 3, 4]);

####
```

A.7 Code Maple™ pour les surfaces de lenteur

```
#####
# Surfaces de lenteur en TTI 3D #
#####

restart:
with(LinearAlgebra):

### paramètres TTI
rho:=1:
Vp:=3000:
Vs:=1500:
epsilon:=0.25:
delta:=0.10:
gam:=0.20:
theta_TTI:=30*Pi/180:
phi_TTI:=15*Pi/180:

### tenseur VTI
Cvti:=Matrix(6,6):
Cvti[1,1]:=rho*Vp^2*(1+2*epsilon):
Cvti[2,2]:=Cvti[1,1]:
Cvti[3,3]:=rho*Vp^2:
Cvti[4,4]:=rho*Vs^2:
Cvti[5,5]:=Cvti[4,4]:
Cvti[6,6]:=rho*Vs^2*(1+2*gam):
Cvti[1,2]:=Cvti[1,1]-2*Cvti[6,6]:
Cvti[1,3]:=sqrt(rho^2*(Vp^2-Vs^2)^2+2*delta*Vp^2*rho^2*(Vp^2-Vs^2)) - rho*Vs^2:
Cvti[2,3]:=Cvti[1,3]:

### TTI par rotation du VTI
R := Matrix([
[cos(theta_TTI)*cos(phi_TTI),cos(theta_TTI)*sin(phi_TTI),-sin(theta_TTI)],
[-sin(phi_TTI),cos(phi_TTI),0],
[sin(theta_TTI)*cos(phi_TTI),sin(theta_TTI)*sin(phi_TTI),cos(theta_TTI)]]):
V := Matrix([[1,6,4],[6,2,5],[4,5,3]]):
for i from 1 to 6 do
  for j from i to 6 do
    Cvti[j,i]:=Cvti[i,j]:
  od;
od;
C:=Matrix(6,6):
for i from 1 to 3 do
  for j from 1 to 3 do
    for k from 1 to 3 do
      for l from 1 to 3 do
```

```

temp:=0:
for p from 1 to 3 do
  for q from 1 to 3 do
    for r from 1 to 3 do
      for s from 1 to 3 do
        temp:=temp+R[p,i]*R[q,j]*R[r,k]*R[s,l]*Cvtti[V[p,q],V[r,s]]:
      od;
    od;
  od;
od;
C[V[i,j],V[k,l]]:=temp:
od;
od;
od;

### équation de dispersion
A:=Matrix(3,3):
A[1,1]:=C[1,1]*x^2+C[6,6]*y^2+C[4,4]*z^2+2*C[1,6]*x*y+2*C[1,4]*x*z+2*C[4,6]*y*z:
A[1,2]:=(C[6,6]+C[1,2])*x*y+C[1,6]*x^2+(C[1,5]+C[4,6])*x*z \
+C[2,6]*y^2+(C[5,6]+C[2,4])*y*z+C[4,5]*z^2:
A[1,3]:=(C[4,4]+C[1,3])*x*z+C[1,4]*x^2+(C[1,5]+C[4,6])*x*y \
+C[5,6]*y^2+(C[3,6]+C[4,5])*y*z+C[3,4]*z^2:
A[2,1]:=A[1,2]:
A[3,1]:=A[1,3]:
A[2,2]:=C[6,6]*x^2+C[2,2]*y^2+C[5,5]*z^2+2*C[2,6]*x*y+2*C[5,6]*x*z+2*C[2,5]*y*z:
A[2,3]:=(C[5,5]+C[2,3])*y*z+C[4,6]*x^2+(C[5,6]+C[2,4])*x*y \
+(C[3,6]+C[4,5])*x*z+C[2,5]*y^2+C[3,5]*z^2:
A[3,2]:=A[2,3]:
A[3,3]:=C[4,4]*x^2+C[5,5]*y^2+C[3,3]*z^2+2*C[4,5]*x*y+2*C[3,4]*x*z+2*C[3,5]*y*z:
eVal:=Eigenvalues(evalf(A)):

### surfaces de lenteur
slow_p:=subs(x=cos(theta)*sin(phi),y=sin(theta)*sin(phi),z=cos(phi),sqrt(rho/eVal[1])):
slow_s1:=subs(x=cos(theta)*sin(phi),y=sin(theta)*sin(phi),z=cos(phi),sqrt(rho/eVal[2])):
slow_s2:=subs(x=cos(theta)*sin(phi),y=sin(theta)*sin(phi),z=cos(phi),sqrt(rho/eVal[3])):
#plot3d(slow_p,theta=0..2*Pi,phi=0..Pi,coords=spherical,scaling=CONSTRAINED)
#plot3d(slow_s1,theta=0..2*Pi,phi=0..Pi,coords=spherical,scaling=CONSTRAINED)
#plot3d(slow_s2,theta=0..2*Pi,phi=0..Pi,coords=spherical,scaling=CONSTRAINED)

### coupes suivant les axes du repère cartésien
slow_p_x:=eval(slow_p,phi=Pi/2):
slow_p_y:=eval(slow_p,theta=0):
slow_p_z:=eval(slow_p,theta=Pi/2):
slow_s1_x:=eval(slow_s1,phi=Pi/2):
slow_s1_y:=eval(slow_s1,theta=0):
slow_s1_z:=eval(slow_s1,theta=Pi/2):

```

A.7. Code Maple™ pour les surfaces de lenteur

```
slow_s2_x:=eval(slow_s2,phi=Pi/2):
slow_s2_y:=eval(slow_s2,theta=0):
slow_s2_z:=eval(slow_s2,theta=Pi/2):
plot([slow_p_x,slow_s1_x,slow_s2_x],theta=0..2*Pi,coords=polar,scaling=CONSTRAINED);
plot([slow_p_y,slow_s1_y,slow_s2_y],phi=0..2*Pi,coords=polar,scaling=CONSTRAINED);
plot([slow_p_z,slow_s1_z,slow_s2_z],phi=0..2*Pi,coords=polar,scaling=CONSTRAINED);

####
```


Bibliographie

- [AAD⁺10] E. Agullo, C. Augonnet, J. Dongarra, H. Ltaief, R. Namyst, S. Thibault, and S. Tomov. Faster, cheaper, better - a hybridization methodology to develop linear algebra software for gpus. In Wen mei W. Hwu, editor, *GPU Computing Gems*, volume 2. Morgan Kaufmann, 2010.
- [AAD⁺11a] E. Agullo, C. Augonnet, J. Dongarra, M. Faverge, J. Langou, H. Ltaief, and S. Tomov. LU factorization for accelerator-based systems. In Howard Jay Siegel and Amr El-Kadi, editors, *The 9th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, pages 217–224, 2011.
- [AAD⁺11b] E. Agullo, C. Augonnet, J. Dongarra, M. Faverge, H. Ltaief, S. Thibault, and S. Tomov. QR Factorization on a Multicore Node Enhanced with Multiple GPU Accelerators. In *IPDPS proceedings*, 2011.
- [ABC⁺14] E. Agullo, B. Bramas, O. Coulaud, E. Darve, M. Messner, and T. Takahashi. Task-based fmm for multicore architectures. *SIAM Journal Scientific Computing*, 36(1) :66–93, 2014.
- [ABCM02] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39 (5) :1749–1779, 2002.
- [ABGL13] E. Agullo, A. Buttari, A. Guermouche, and F. Lopez. Multifrontal QR factorization for multicore architectures over runtime systems. In Felix Wolf, Bernd Mohr, and Dieter Mey, editors, *Euro-Par 2013 Parallel Processing*, volume 8097 of *Lecture Notes in Computer Science*, pages 521–532. Springer Berlin Heidelberg, 2013.
- [AGG⁺12] E. Agullo, L. Giraud, A. Guermouche, S. Nakov, and J. Roman. Task-based conjugate-gradient for multi-gpus platforms. Rapport de recherche RR-8192, INRIA, 2012.
- [Alk98] T. Alkhalifah. Acoustic approximations for seismic processing in transversely isotropic media. *Geophysics*, 63 :623–631, 1998.
- [Alk00] T. Alkhalifah. An acoustic wave equation for anisotropic media. *Geophysics*, 65 :1239–1250, 2000.
- [AMM06] M. Ainsworth, P. Monk, and W. Muniz. Dispersive and dissipative properties of discontinuous galerkin finite element methods for the second-order wave equation. *Journal of Scientific Computing*, 27 (1-3) :–, 2006.
- [AR02] K. Aki and P. G. Richards. *Quantitative Seismology, 2nd edition*. University science books, 2002.

Bibliographie

- [ATNW11] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. StarPU : A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. *Concurrency and Computation : Practice and Experience, Special Issue : Euro-Par 2009*, 23 :187–198, 2011.
- [Aug11] C. Augonnet. *Scheduling Tasks over Multicore machines enhanced with Accelerators : a Runtime System's Perspective*. PhD thesis, Bordeaux 1 university, France, 2011.
- [Bal09] C. Baldassari. *Modélisation et simulation numérique pour la migration terrestre par équation d'ondes*. PhD thesis, Université de Pau et des Pays de l'Adour, 2009.
- [BBAC14] L. Boillot, G. Bosilca, E. Agullo, and H. Calandra. Task-based programming for seismic imaging : Preliminary results. In *Proceedings of the 16th IEEE International Conference on High Performance Computing and Communications (HPCC)*, pages 1259–1266, 2014.
- [BBC⁺10] Z. Budimlić, M. Burke, V. Cavé, K. Knobe, G. Lowney, R. Newton, J. Palsberg, D. Peixotto, V. Sarkar, F. Schlimbach, and S. Taşirlar. Concurrent collections. *Sci. Program.*, 18(3-4) :203–217, 2010.
- [BBCD14] H. Barucq, L. Boillot, H. Calandra, and J. Diaz. Absorbing boundary conditions for 2d tilted transverse isotropic media. In EDP Sciences and SMAI, editors, *ESAIM : Proceedings*, volume 45 of *Congrès SMAI 2013*, pages 400–409, 2014.
- [BBD⁺11a] G. Bosilca, A. Bouteiller, A. Danalis, M. Faverge, A. Haidar, T. Hérault, J. Kurzak, J. Langou, P. Lemarinier, H. Ltaief, P. Luszczek, A. YarKhan, and J. Dongarra. Flexible Development of Dense Linear Algebra Algorithms on Massively Parallel Architectures with DPLASMA. In *IPDPS Workshops*, pages 1432–1441, 2011.
- [BBD⁺11b] G. Bosilca, A. Bouteiller, A. Danalis, M. Faverge, A. Haidar, T. Hérault, J. Kurzak, J. Langou, P. Lemarinier, H. Ltaief, P. Luszczek, A. YarKhan, and J. Dongarra. Flexible development of dense linear algebra algorithms on massively parallel architectures with DPLASMA. In *12th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC'11)*, 2011.
- [BBD⁺12] G. Bosilca, A. Bouteiller, A. Danalis, T. Hérault, P. Lemarinier, and J. Dongarra. DAGuE : A generic distributed DAG engine for high performance computing. *Parallel Computing*, 38 :37–51, 2012.
- [BBD⁺13] G. Bosilca, A. Bouteiller, A. Danalis, M. Faverge, T. Hérault, and J.J. Dongarra. PaRSEC : Exploiting Heterogeneity to Enhance Scalability. *Computing in Science Engineering*, 15(6) :36–45, 2013.
- [BCC⁺06] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, et al. Grid'5000 : a large scale and highly reconfigurable experimental grid testbed. *International Journal of High Performance Computing Applications*, 20(4) :481–494, 2006.
- [BCOM⁺10] F. Broquedis, J. Clet-Ortega, S. Moreaud, N. Furmento, B. Goglin, G. Mercier, S. Thibault, and R. Namyst. hwloc : a generic framework for managing

- hardware affinities in HPC applications. In *Proceedings of the 18th Euro-micro International Conference on Parallel, Distributed and Network-Based Processing (PDP2010)*, 2010.
- [Bec91] E. Becache. *Résolution par une méthode d'équations intégrales d'un problème de diffraction d'ondes élastiques transitoires par une fissure*. PhD thesis, Paris VI, 1991.
- [Ber66] A. J. Bernstein. Analysis of programs for parallel processing. *IEEE Transactions on Electronic Computers*, 15(5) :757–763, 1966.
- [Ber94] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114 :185–200, 1994.
- [Ber96] J.-P. Berenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J. of Comp. Phys.*, 127 :363–379, 1996.
- [BFJ03] E. Bécache, S. Fauqueux, and P. Joly. Stability of perfectly matched layers, group velocities and anisotropic waves. *Journal of Computational Physics*, 188(2) :399–433, 2003.
- [BGCDL14] M. Bonnasse-Gahot, H. Calandra, J. Diaz, and S. Lanteri. Discontinuous galerkin methods for solving helmholtz elastic wave equations for seismic imaging. In *ECCOMAS - Book of abstracts*, 2014.
- [BGH10] E. Bécache, D. Givoli, and T. Hagstrom. High-order absorbing boundary conditions for anisotropic and convective wave equations. *Journal of Computational Physics*, 229 :1099–1129, 2010.
- [BJK⁺96] R.D. Blumofe, C. F. Joerg, B. C. Kuszmaul, Charles E. Leiserson, Keith H. Randall, and Yuli Zhou. Cilk : An efficient multithreaded runtime system. *Journal of parallel and distributed computing*, 37(1) :55–69, 1996.
- [BKS83] E. Baysal, D.D. Kosloff, and J.W.C. Sherwood. Reverse time migration. *Geophysics*, 48(11) :1514–1524, 1983.
- [BLKD08] A. Buttari, J. Langou, J. Kurzak, and J. Dongarra. Parallel tiled QR factorization for multicore architectures. *Concurrency and Computation : Practice and Experience*, 20(13) :1573–1590, 2008.
- [BLP06] M. Bernacki, S. Lanteri, and S. Piperno. Time-domain parallel simulation of heterogeneous wave propagation on unstructured grids using explicit, non-diffusive, discontinuous galerkin methods. *J. Comput. Acoust.*, 14 (1) :57–81, 2006.
- [BSW08] J. D. De Basade, M. K. Sen, and M. F. Wheeler. The interior penalty discontinuous galerkin method for elastic wave propagation : grid dispersion. *Geophys. J. Int.*, 175 :83–93, 2008.
- [BT80] A. Bayliss and E. Turkel. Radiation boundary conditions for wave-like equations. *Comm. Pure Appl. Math.*, 33 :707–725, 1980.
- [CD06] E. Caron and F. Desprez. Diet : A scalable toolbox to build network enabled servers on the grid. *International Journal of High Performance Computing Applications*, 20(3) :335–352, 2006.
- [CGL09] B. Cockburn, J. Gopalakrishnan, and R. Lazarov. Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for

Bibliographie

- second order elliptic problems. *SIAM Journal on Numerical Analysis*, 47 :1319–1365, 2009.
- [CJ99] M. Cosnard and E. Jeannot. Compact DAG representation and its dynamic scheduling. *Journal of Parallel and Distributed Computing*, 58(3) :487–514, 1999.
- [CKS00a] B. Cockburn, G. Karniadakis, and C. Shu. *Discontinuous Galerkin methods, theory, computation and applications*, volume 11. Springer, Berlin, 2000.
- [CKS00b] B. Cockburn, G.E. Karniadakis, and C.-W. Shu. *The development of discontinuous Galerkin methods*, volume 11 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, New York, 2000.
- [Cla71] J.F. Claerbout. Toward a unified theory of reflector mapping. *Geophysics*, 36(3) :467–481, 1971.
- [Coc01] B. Cockburn. Devising discontinuous galerkin methods for non-linear hyperbolic conservation laws. *J. Comput. Appl. Math.*, 128 :187–204, 2001.
- [Col93] F. Collino. *High order absorbing boundary conditions for wave propagations model : Straight line boundary and corner cases*. SIAM, 1993.
- [CQQv07] E. Chan, E. S. Quintana-Orti, G. Gregorio Quintana-Orti, and R. van de Geijn. Supermatrix out-of-order scheduling of matrix operations for smp and multi-core architectures. In *Nineteenth Annual ACM Symposium on Parallel Algorithms and Architectures SPAA '07*, pages 116–125, 2007.
- [CT01] F Collino and C. Tsogka. Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*, 66(1) :294–307, 2001.
- [DB11] E. Duveneck and M. P. Bakker. Stable p-wave modeling for reverse-time migration in tilted TI media. *Geophysics*, 76(2) :65–75, 2011.
- [DBB⁺14] A. Danalis, G. Bosilca, A. Bouteiller, T. Herault, and J. Dongarra. PTG : An abstraction for unhindered parallelism. In *Proceedings of the Fourth International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing, WOLFHPC '14*, pages 21–30, Piscataway, NJ, USA, 2014. IEEE Press.
- [DBea11] J. Dongarra and P. Beckman et al. The international exascale software roadmap. *International Journal of High Performance Computer Applications*, 25(1), 2011.
- [DBL07] X. Du, J. Brancroft, and L. Lines. Anisotropic reverse-time migration for tilted ti media. *Geophysical Prospecting*, 55(6) :853–869, 2007.
- [DDG⁺12] J. Dongarra, T. Dong, M. Gates, A. Haidar, S. Tomov, and I. Yamazaki. MAGMA ; a new generation of linear algebra libraries for GPU and multicore architecture, 2012. SuperComputing conference.
- [DE90] J. Dellinger and J. Etgen. Wave-field separation in two-dimensional anisotropic media. *Geophysics*, 55 :914–919, 1990.
- [Den74] J. B. Dennis. First version of a data flow procedure language. In *Programming Symposium*, pages 362–376. Springer Berlin Heidelberg, 1974.

- [DFF08] X. Du, R. P. Fletcher, and P. Fowler. A new pseudo-acoustic wave equation for VTI media. In *EAGE*, page H033, 2008.
- [DFGO09] S. Delcourte, L. Fezoui, and N. Glinsky-Olivier. A high-order discontinuous galerkin method for the seismic wave propagation. *ESAIM : Proceedings*, 27 :70–89, 2009. CANUM 2008.
- [DH13] J. Dongarra and M.A. Heroux. Toward a new metric for ranking high performance computing systems. Technical report, Sandia National Laboratories, 2013.
- [DK06] M. Dumbser and M. Käser. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes II : the three-dimensional isotropic case. *Geophysical J. Int.*, 167 :319–336, 2006.
- [DM98] L. Dagum and R. Menon. OpenMP : an industry standard API for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1) :46–55, 1998.
- [DMBP08] E. Duveneck, P. Milcik, M. P. Bakker, and C. Perkins. Acoustic VTI wave equations and their application for anisotropic reverse-time migration. In *SEG*, pages 2186–2190, 2008.
- [DMS97] J. Dongarra, H.W. Meuer, and E. Strohmaier. TOP500 supercomputer sites. *Supercomputer*, 13 :89–111, 1997.
- [Don88] J. Dongarra. The LINPACK benchmark : An explanation. In *Supercomputing*, pages 456–474. Springer Berlin Heidelberg, 1988.
- [DPA⁺08] A. Duran, J. M. Perez, R. M. Ayguadé, E. amd Badia, and J. Labarta. Extending the OpenMP tasking model to allow dependent tasks. In *OpenMP in a New Era of Parallelism, 4th International Workshop, IWOMP*, volume 5004 of *Lecture Notes in Computer Science*, pages 111–122, 2008.
- [EM77] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.*, 31 :629–651, 1977.
- [EM79] B. Engquist and A. Majda. Radiation boundary conditions for acoustic and elastic wave calculations. *Comm. Pure Appl. Math.*, 32 :314–358, 1979.
- [FDF09] R. P. Fletcher, X. Du, and P. Fowler. Reverse time migration in tilted transversely isotropic (TTI) media. *Geophysics*, 74(6) :179–187, 2009.
- [FDF10] P. Fowler, X. Du, and R. P. Fletcher. Coupled equations for reverse time migration in transversely isotropic media. *Geophysics*, 75(1) :11–22, 2010.
- [GBP07] T. Gautier, X. Besseron, and L. Pigeon. KAAPI : A thread scheduling runtime system for data flow computations on cluster of multi-processors. In *Proceedings of the 2007 international workshop on Parallel symbolic computation, ACM*, pages 15–23, 2007.
- [GDDW11] H. Guan, E. Dussaud, B. Denel, and P. Williamson. Techniques for an efficient implementation of RTM in TTI media. In *SEG Annual Meeting*, pages 3393–3397, 2011.
- [Giv03] D. Givoli. *Exact and high order non-reflecting computational boundaries*, volume Mathematical and Numerical Aspects of Wave Propagation. G. Cohen, E. Heikkola, P. Joly and P. Neittanmaki, Eds., 2003.

Bibliographie

- [GK95] M. J. Grote and J. B. Keller. Exact nonreflecting boundary conditions for the time dependent wave equation. *SIAM J. Appl. Math.*, 55 :280–297, 1995.
- [GLFR13] T. Gautier, F. Lementec, V. Faucher, and B. Raffin. X-Kaapi : a multi paradigm runtime for multicore architectures. In *Parallel Processing (ICPP), 42nd International Conference, IEEE*, pages 728–735, 2013.
- [GSS06] M. J. Grote, A. Schneebeli, and D. Schoötzau. Discontinuous galerkin finite element method for the wave equation. *SIAM J. Numer. Anal.*, 44 (6) :2408–2431, 2006.
- [GZR04] V. Grechka, L. Zhang, and J. Rector. Shear waves in acoustic anisotropic media. *Geophysics*, 69(2) :576–582, 2004.
- [Hag03] T. Hagström. *New results on absorbing layers and radiation boundary conditions*, volume Topics in Computational Wave Propagation - Direct and Inverse Problems of *Computational Science and Engineering*. M. Ainsworth, P. Davies, D. Duncan, P. Martin and B. Rynne, Eds, 2003.
- [Hal80] L. Halpern. *Etude de conditions aux limites absorbantes pour des schemas numeriques relatifs a des equations hyperboliques lineaires*. PhD thesis, Paris VI, 1980.
- [Hig91] R. L. Higdon. Absorbing boundary conditions for elastic waves. *Geophysics*, 56 (2) :231–241, 1991.
- [HPBR11] L. Halpern, S Petit-Bergez, and J. Rauch. The analysis of matched layers. *Confluentes Math.*, 3(2) :159–236, 2011.
- [HZZY09] T. Huang, Y. Zhang, H. Zhang, and J. Young. Subsalt imaging using tti reverse time migration. *The leading edge*, pages 448–451, 2009.
- [JJR10] S. Jin, F. Jiang, and Y. Ren. Comparison of isotropic, vti and tti reverse time migration : an experiment on bp anisotropic benchmark dataset. In *SEG*, 2010.
- [JNP84] C. Johnson, U. Nävert, and J. Pitkänta. Finite element methods for linear hyperbolic equations. *Comput. Methods Appl. Mech. Engrg.*, 45 :285–312, 1984.
- [JP86] C. Johnson and J. Pitkänta. An analysis of the discontinuous galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46 :1–26, 1986.
- [JR13] J. Jeffers and J. Reinders. *Intel Xeon Phi Coprocessor High-Performance Programming*. ELSEVIER, 2013.
- [KD06] M. Käser and M. Dumbser. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes I : The two-dimensional isotropic case with external source terms. *Geophysical J. Int.*, 166 (2) :855–877, 2006.
- [KHdlP08] M. Käser, V. Hermann, and J. de la Puente. Quantitative accuracy analysis of discontinuous galerkin method for seismic wave propagation. *Geophys. J. Int.*, 173 :990–999, 2008.
- [KLY⁺13] J. Kurzak, P. Luszczek, A. YarKhan, M. Faverge, J. Langou, H. Bouwmeester, and J. Dongarra. *Multithreading in the PLASMA library*, chapter 5, pages 119–141. Multi and Many-core processing : architecture, programming, algorithms, & Applications. Taylor & Francis, 2013.

-
- [Kri14] R. Kriemann. *H*-lu factorization on many-core systems, 2014. Preprint 5.
 - [LBZN09] W. Liu, K.P. Bube, L.B. Zhang, and K. Nihei. Stable reverse-time migration in variable-tilt TI media. In *EAGE*, page P064, 2009.
 - [Lei10] C. E. Leiserson. The cilk++ concurrency platform. *The Journal of Supercomputing*, 51(3) :244–257, 2010.
 - [Les75] P. Lesaint. *Sur la résolution des systèmes hyperboliques du premier ordre par des methodes d’éléments finis*. PhD thesis, Université Paris 6, 1975.
 - [LFR⁺14] X. Lacoste, M. Faverge, P. Ramet, S. Thibault, and G. Bosilca. Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes. In *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing Workshops and Phd Forum (IPDPSW’14), HCW 2014*, 2014.
 - [LHZAP⁺08] A.C. Lesage, H. H. Zhou, M. Araya-Polo, J.M. Cela, and F. Ortigosa. Hybrid finite difference-pseudospectral method for 3d rtm in tti media. In *EAGE*, 2008.
 - [LMJ⁺09] F. Liu, S.A. Morton, L. Jiang, L. Ni, and J.P. Leveille. Decoupled p and SV waves in an acoustic VTI media. In *79st Annual International Meeting SEG Expanded Abstracts*, volume 28, pages 2844–2848, 2009.
 - [LR74] P. Lesaint and P.-A. Raviart. *On a finite element method for solving the neutron transport equation*. Academic Press, New York, 1974.
 - [LSAT13] B. Lize, G. Sylvand, E. Agullo, and S. Thibault. A task-based h-matrix solver for acoustic and electromagnetic problems on multicore architectures. In *SciCADE, the International Conference on Scientific Computation and Differential Equations*, 2013.
 - [LY14] H. Ltaief and R. Yokota. Data-driven execution of fast multipole methods. *Concurrency and Computation : Practice and Experience*, 26(11) :1935–1946, 2014.
 - [Maple] MaplesoftTM. Maple, <http://www.maplesoft.com/products/maple/>.
 - [MBL⁺14] L. Métivier, R. Brossier, S. Labbé, S. Operto, and J. Virieux. A robust absorbing layer method for anisotropic seismic wave modeling. *Journal of Computational Physics*, 279 :218–240, 2014.
 - [McM83] G.A. McMechan. Migration by extrapolation of time-dependent boundary values. *Geophysical Prospecting*, 31(3) :413–420, 1983.
 - [Moo65] G.E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8) :1–4, 1965.
 - [MPI98] *MPI—the Complete Reference : The MPI core*, volume 1. Snir, Marc, 1998.
 - [Näv82] U. Nävert. *A Finite Element Method for Convection-Diffusion Problems*. PhD thesis, Chalmers University of Technology, 1982.
 - [Ope11] The openacc application programming interface - specification, 2011.
 - [PBL07] P.M. Perez, R.M. Badia, and J. Labarta. A flexible and portable programming model for SMP and multi-cores. Technical report, BSC-UPC, 2007.
 - [Pod12] O. Podgornova. Transparent boundary conditions for elastic anisotropic (VTI) media : axially symmetric case. *Commun. Comput. Phys.*, 11(2) :541–554, 2012.

Bibliographie

- [PUS11] R. Pestana, B. Ursin, and P. Stoffa. Separated p- and SV-wave equations for VTI media. In *81st Annual International Meeting SEG Expanded Abstracts*, volume 30, pages 163–167, 2011.
- [QOIQOvdG09] G. Quintana-Ortí, F. D. Igual, E. S. Quintana-Ortí, and R. A. van de Geijn. Solving dense linear systems on platforms with multiple hardware accelerators. *ACM SIGPLAN Notices*, 44(4) :121–130, 2009.
- [QOQOC⁺08] G. Quintana-Ortí, E. S. Quintana-Ortí, E. Chan, F. G. Van Zee, and R. A. van de Geijn. Scheduling of QR factorization algorithms on SMP and multi-core architectures. In *Proceedings of PDP’08*, FLAME Working Note #24, 2008.
- [Rei07] J. Reinders. *Intel threading building blocks : outfitting C++ for multi-core processor parallelism*. O’Reilly Media Inc., 2007.
- [Rey78] A. C. Reynolds. Boundary conditions for the numerical solution of wave equations problem. *Geophysics*, 43 (6) :1099–1110, 1978.
- [RH73] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report la-ur-73-479, Los Alamos Scientific Laboratory, 1973.
- [RHAT13] C. Rossignon, P. Henon, O. Aumage, and S. Thibault. A NUMA-aware fine grain parallelization framework for multi-core architecture. In *IPDPS Proc.*, 2013.
- [Ric53] N. Ricker. The form and laws of propagation of seismic wavelets. *Geophysics*, 18 :10–40, 1953.
- [Sar89] V. Sarkar. *Partitioning and scheduling parallel programs for multiprocessors*. MIT press, 1989.
- [SG10a] S. Savadatti and M.N. Guddati. Absorbing boundary conditions for scalar waves in anisotropic media. part 1 : Time harmonic modeling. *J. Comput. Phys.*, 229(19) :6696–6714, 2010.
- [SG10b] S. Savadatti and M.N. Guddati. Absorbing boundary conditions for scalar waves in anisotropic media. part 2 : Time-dependent modeling. *J. Comput. Phys.*, 229(18) :6644–6662, 2010.
- [SG12a] S. Savadatti and M.N. Guddati. Accurate absorbing boundary conditions for anisotropic elastic media. part 1 : Elliptic anisotropy. *J. Comput. Phys.*, 231(22) :7584–7607, 2012.
- [SG12b] S. Savadatti and M.N. Guddati. Accurate absorbing boundary conditions for anisotropic elastic media. part 2 : Untilted non-elliptic anisotropy. *J. Comput. Phys.*, 231(22) :7608–7625, 2012.
- [SHS00] E. Süli, P. Houston, and C. Schwab. hp-finite element methods for hyperbolic problems. In *MAFELAP X*, pages 143–162, 2000.
- [SSH00] E. Süli, C. Schwab, and P. Houston. *hp-DGFEM for partial differential equations with non-negative characteristic form*, volume 11 of *Lect. Notes Comput. Sci. Eng.* Springer Berlin, 2000.
- [Sut05] H. Sutter. The free lunch is over : A fundamental turn toward concurrency in software. *Dr. Dobbs’s Journal*, 30(3), 2005.

-
- [Sut12] H. Sutter. Welcome to the jungle. *http://herbsutter.com*, 2012.
 - [Tay81] M. Taylor. *Pseudodifferential operators*. Princeton, 1981.
 - [Tho86] L. Thomsen. Weak elastic anisotropy. *Geophysics*, 51(10) :1954–1966, 1986.
 - [Tso99] C. Tsogka. *Modélisation mathématique et numérique de la propagation des ondes élastiques tridimensionnelles dans des milieux fissurés*. PhD thesis, Paris IX Dauphine, 1999.
 - [Tsv96] I. Tsvankin. P-wave signatures and notation for transversely isotropic media : An overview. *Geophysics*, 61(2) :467–483, 1996.
 - [Tsv01] I. Tsvankin. *Seismic signatures and analysis of reflection data in anisotropic media*. Elsevier, 2001.
 - [TT94] I. Tsvankin and L. Thomsen. Nonhyperbolic reflection moveout in anisotropic media. *Geophysics*, 59(8) :1290–1304, 1994.
 - [Ven14] F. Ventimiglia. *Schémas numériques d’ordre élevé en temps et en espace pour l’équation des ondes du premier ordre. Application à la Reverse Time Migration*. PhD thesis, Université de Pau et des Pays de L’Adour, 2014.
 - [Wen14] G. Wenes. Exascale computing for seismic exploration, 2014. CRAY blog.
 - [Whi83] N.D. Whitmore. Iterative depth migration by backward time propagation. In *SEG*, pages 827–830, 1983.
 - [YGW14] P. Yanga, J. Gaoa, and B. Wangc. RTM using effective boundary saving : A staggered grid GPU implementation. *Computers & Geosciences*, 68 :64–72, 2014.
 - [YKD11] A. YarKhan, J. Kurzak, and J. Dongarra. QUARK users’ guide : QUeueing And Runtime for Kernels. Technical Report ICL-UT-11-02, Innovative Computing Laboratory, University of Tennessee, 2011.
 - [YS08] J. Yan and P. Sava. Isotropic angle-domain elastic reverse-time migration. *Geophysics*, 73(6) :229–239, 2008.
 - [YS09] J. Yan and P. Sava. Elastic wave-mode separation for VTI media. *Geophysics*, 74 :19–32, 2009.
 - [YS11] J. Yan and P. Sava. Improving the efficiency of elastic wave-mode separation for heterogeneous tilted transverse isotropic media. *Geophysics*, 76 :65–78, 2011.
 - [ZCvdG⁺09] F. G. Van Zee, E. Chan, R. A. van de Geijn, E. S. Quintana-Orti, and G. Quintana-Orti. The `libflame` Library for Dense Matrix Computations. *Computing in Science and Engineering*, 11(6) :56–63, 2009.
 - [ZLNM⁺09] H. Zhu, Y. Luo, T. Nissen-Meyer, C. Morency, and J. Tromp. Elastic imaging and time-lapse migration based on adjoint methods. *Geophysics*, 74(6) :WCA167–WCA177, 2009.
 - [ZPS12] G. Zhan, R.C. Pestana, and P.L. Stoffa. Decoupled equations for reverse time migration in tilted transversely isotropic media. *Geophysics*, 77(2) :37–45, 2012.
 - [ZZ08] H. Zhang and Y. Zhang. Reverse time migration in 3D heterogeneous TTI media. In *SEG*, pages 2196–2200, 2008.

Bibliographie

- [ZZ09] Y. Zhang and H. Zhang. A stable TTI reverse time migration and its implementation. In *SEG*, pages 2794–2798, 2009.
- [ZZB06a] H. Zhou, G. Zhang, and R. Bloor. An anisotropic acoustic wave equation for modeling and migration in 2d tti media. In *SEG*, 2006.
- [ZZB06b] H. Zhou, G. Zhang, and R. Bloor. An anisotropic acoustic wave equation for vti media. In *EAGE*, 2006.
- [ZZZ11] Y. Zhang, H. Zhang, and G. Zhang. A stable TTI reverse time migration and its implementation. *Geophysics*, 76(3) :3–11, 2011.

Contributions à la modélisation mathématique et à l’algorithmique parallèle pour l’optimisation d’un propagateur d’ondes élastiques en milieu anisotrope

Résumé : La méthode d’imagerie la plus répandue dans l’industrie pétrolière est la RTM (*Reverse Time Migration*) qui repose sur la simulation de la propagation des ondes dans le sous-sol. Nous nous sommes concentrés sur un propagateur d’ondes élastiques 3D en milieu anisotrope de type TTI (*Tilted Transverse Isotropic*). Nous avons directement travaillé dans le code de recherche de Total DIVA (*Depth Imaging Velocity Analysis*), basé sur une discrétisation par la méthode de Galerkin Discontinue et le schéma Leap-Frog, et développé pour le calcul parallèle intensif – HPC (*High Performance Computing*). Nous avons ciblé plus particulièrement deux contributions possibles qui, si elles supposent des compétences très différentes, ont la même finalité : réduire les coûts de calculs requis pour la simulation. D’une part, les conditions aux limites classiques de type PML (*Perfectly Matched Layers*) ne sont pas stables dans des milieux TTI. Nous avons proposé de formuler une CLA (Conditions aux Limites Absorbantes) stable dans des milieux anisotropes. La méthode de construction repose sur les propriétés des courbes de lenteur, ce qui donne à notre approche un caractère original. D’autre part, le parallélisme initial, basé sur une décomposition de domaine et des communications par passage de messages à l’aide de la bibliothèque MPI, conduit à un déséquilibre de charge qui détériore son efficacité parallèle. Nous avons corrigé cela en remplaçant le paradigme parallélisme par l’utilisation de la programmation à base de tâches sur support d’exécution.

Cette thèse a été réalisée dans le cadre de l’action de recherche DIP (*Depth Imaging Partnership*) qui lie la compagnie pétrolière Total et Inria.

Mots-clés : équation des ondes élastiques, anisotropie TTI (*Tilted Transverse Isotropy*), Conditions aux Limites Absorbantes, programmation parallèle à base de tâches, HPC (*High-Performance Computing*)

Contributions to the mathematical modeling and to the parallel algorithmic for the optimization of an elastic wave propagator in anisotropic media

Abstract: *The most common method of Seismic Imaging is the RTM (Reverse Time Migration) which depends on wave propagation simulations in the subsurface. We focused on a 3D elastic wave propagator in anisotropic media, more precisely TTI (Tilted Transverse Isotropic). We directly worked in the Total code DIVA (Depth Imaging Velocity Analysis) which is based on a discretization by the Discontinuous Galerkin method and the Leap-Frog scheme, and developed for intensive parallel computing – HPC (High Performance Computing). We choose to especially target two contributions. Although they required very different skills, they share the same goal: to reduce the computational cost of the simulation. On one hand, classical boundary conditions like PML (Perfectly Matched Layers) are unstable in TTI media. We have proposed a formulation of a stable ABC (Absorbing Boundary Condition) in anisotropic media. The technique is based on slowness curve properties, giving to our approach an original side. On the other hand, the initial parallelism, which is based on a domain decomposition and communications by message passing through the MPI library, leads to load-imbalance and so poor parallel efficiency. We have fixed this issue by replacing the paradigm for parallelism by the use of task-based programming through runtime system.*

This PhD thesis have been done in the framework of the research action DIP (Depth Imaging Partnership) between the Total oil company and Inria.

Keywords: *elastic wave equation, TTI (Tilted Transverse Isotropy) anisotropy, Absorbing Boundary Conditions, task-based parallel programming, HPC (High-Performance Computing)*